The Construction of Images using Multi-view Affine Relationships

Dawn Kennedy

A Thesis submitted for the Degree of Doctor of Philosophy

UNIVERSITY OF LONDON

Centre for Advanced Instrumentation Systems
University College London

November 2004

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

In this thesis we give a method of synthesising novel views using only existing views and a set of corresponding points across those views. The method described is based on the assumption that the images have been obtained under affine imaging conditions. This has the advantage that the mathematics needed is simpler than in the perspective case but still allows us to synthesise realistic views. We explore the multi-view relationships that exist between corresponding points in three affine images. It is shown that, by careful consideration of where the measurement errors occur, it is possible to improve the accuracy of these multi-view relationships by solving a total least squares problem. We demonstrate how the affine multi-view relationships can be used to encode and accurately reconstruct existing views as a linear combination of two (or more) basis views.

In order to synthesise completely novel views it is necessary to parameterise either the multi-view relationships or the positions of the control points in each of the views. We describe a method that can be used to parameterise either the control points or the co-efficients of the affine multi-view relationships. The parameterisation method uses a set of sample views to parameterise the control points (or the multi-view relationships) in terms of two parameters. These two parameters can then be varied and used to determine the positions of the control points in novel views. Once the positions of the control points in the novel view are known it is possible to render the image using a combination of the intensities in each of the basis views.

The method used to parameterise the existing views allows us to both interpolate between the views and also to extrapolate away from the views. We would expect the quality of the synthesised view to deteriorate as the novel view is extrapolated away from the original set of images. In this thesis we discuss the possibility of using the structure within the total least squares solution combined with the parameterisation described above to determine a limit of extrapolation.

Acknowledgements

Firstly I would like to say thank you to Robert for all his support and encouragement. I could not and would not have done it without you.

Thank you to John and Bernard for all your help, advice, ideas and suggestions, and most of all for all your time and patience over the past few years.

Lastly I say a general thank you to all my family and friends and those of you at Sira who have given me any words of encouragement over the years. Thanks for keeping me smiling.

Contents

List of Figures and Tables

Chapter 1

Introduction

This thesis is about synthesising novel views from a collection of images. In this chapter we introduce the problem of view synthesis. The background and motivation for studying view synthesis is discussed and we give an outline of the solution that will be developed throughout the thesis.

1.1 The Problem

We present a method of view synthesis based on the assumption of affine imaging. We use the affine camera as an approximation to the perspective case. It will be shown later that making this assumption has few practical limitations whilst it considerably simplifies the mathematics and leads to an eigenproblem [Wil65] that can be solved using singular value decomposition [GV96, VV91]. The aim is to develop a simple, robust method of creating perceptually compelling images rather than achieve great quantitative accuracy. No camera calibration assumptions are made so the only information that we use is that which can be extracted from the images used in the synthesis. In this introductory chapter we give the motivation behind the problem and describe the approach that will be developed throughout the thesis. We give an outline of the remaining chapters and highlight the contributions that this thesis makes.

1.1.1 Motivation

View synthesis is the problem of generating new images from old ones. It is a topic that has received a lot of attention over recent years [AS98, BSG98, KB99, KB98, L-H81, PH99, PPHL98, SD95]. One obvious method of synthesising a novel view is to

construct a 3D model of the scene which can then be imaged from various viewpoints. The construction of a 3D model comes from the science of photogrammetry. Photogrammetrists analyse images and image sequences, that are taken with calibrated cameras, in order to determine the size and shape of objects in the scene [Atk96], and use this information to construct a 3D model. This is related to work in computer vision on stereo reconstruction [SS99] where the relative positions of the cameras are used to compute depth maps of the scene. This can be done using calibrated or uncalibrated cameras. In the case where the cameras are calibrated the depth information obtained can be used to determine the 3D locations of points in the scene and hence construct a 3D model. If the cameras are uncalibrated we are able to determine the relative positions of points in the scene and reconstruct the scene up to a perspective transformation. Unfortunately, the construction techniques used are often error sensitive, time consuming and may involve the need to carefully calibrate the cameras.

When we talk about view synthesis within the context of computer vision we usually wish, however, to avoid the need to reconstruct a model. Instead, in view synthesis the aim is to go directly from a limited set of images to a synthesised view, thereby avoiding the need for any explicit model construction. Some examples of synthesises views are shown in figure 1.1. These views have been synthesised using a set of sample images, shown in figure 1.2. In addition, building the best 3D model from the images available and then using the model to generate novel views does not necessarily lead to the best synthesised novel image. Starting from a set of images and using them to go straight to a novel view, without first generating a model, can produce more accurate results as the overall process can be optimised specifically for this purpose and it avoids the error sensitive techniques required in constructing a model.

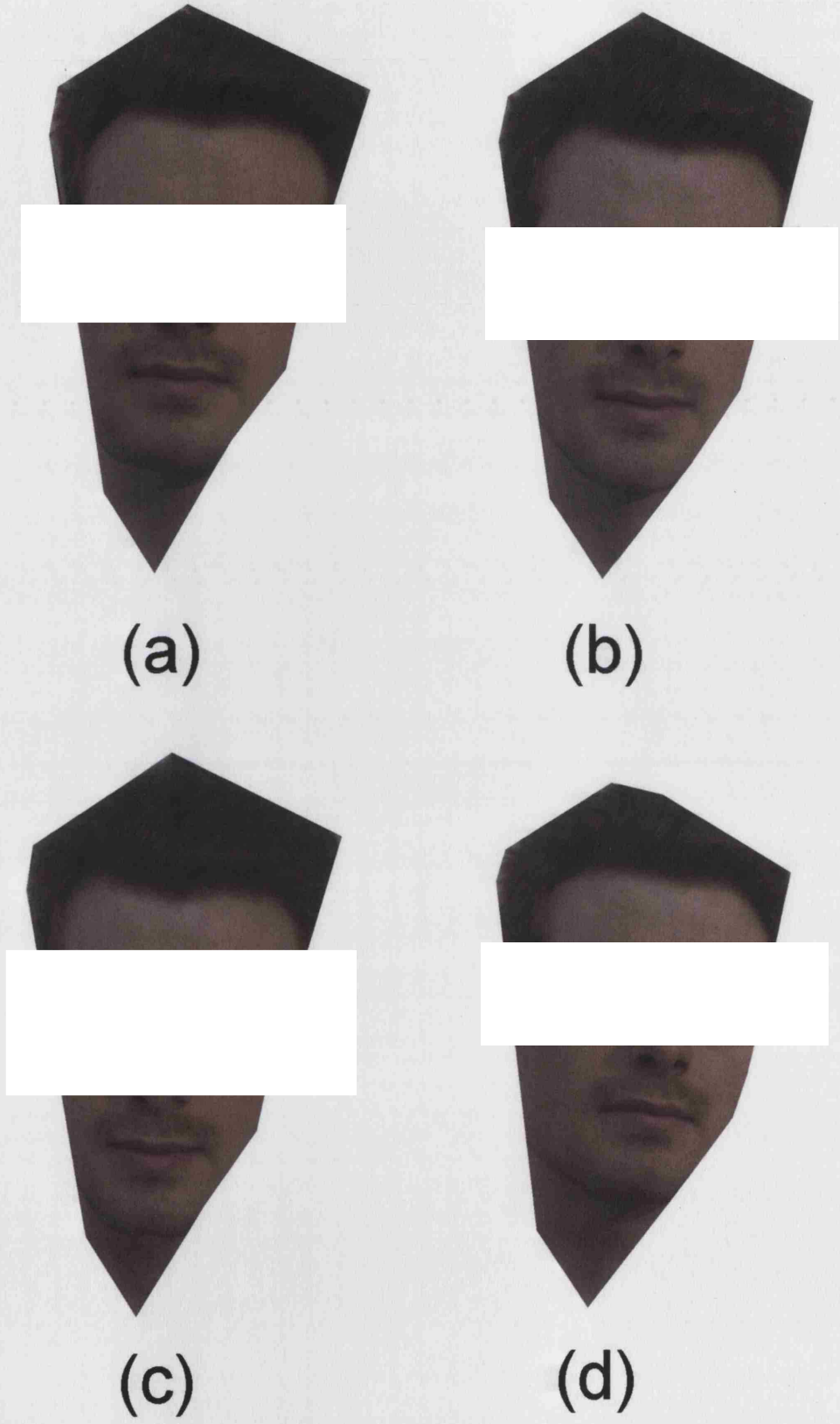


Figure 1.1. Synthesised novel views.

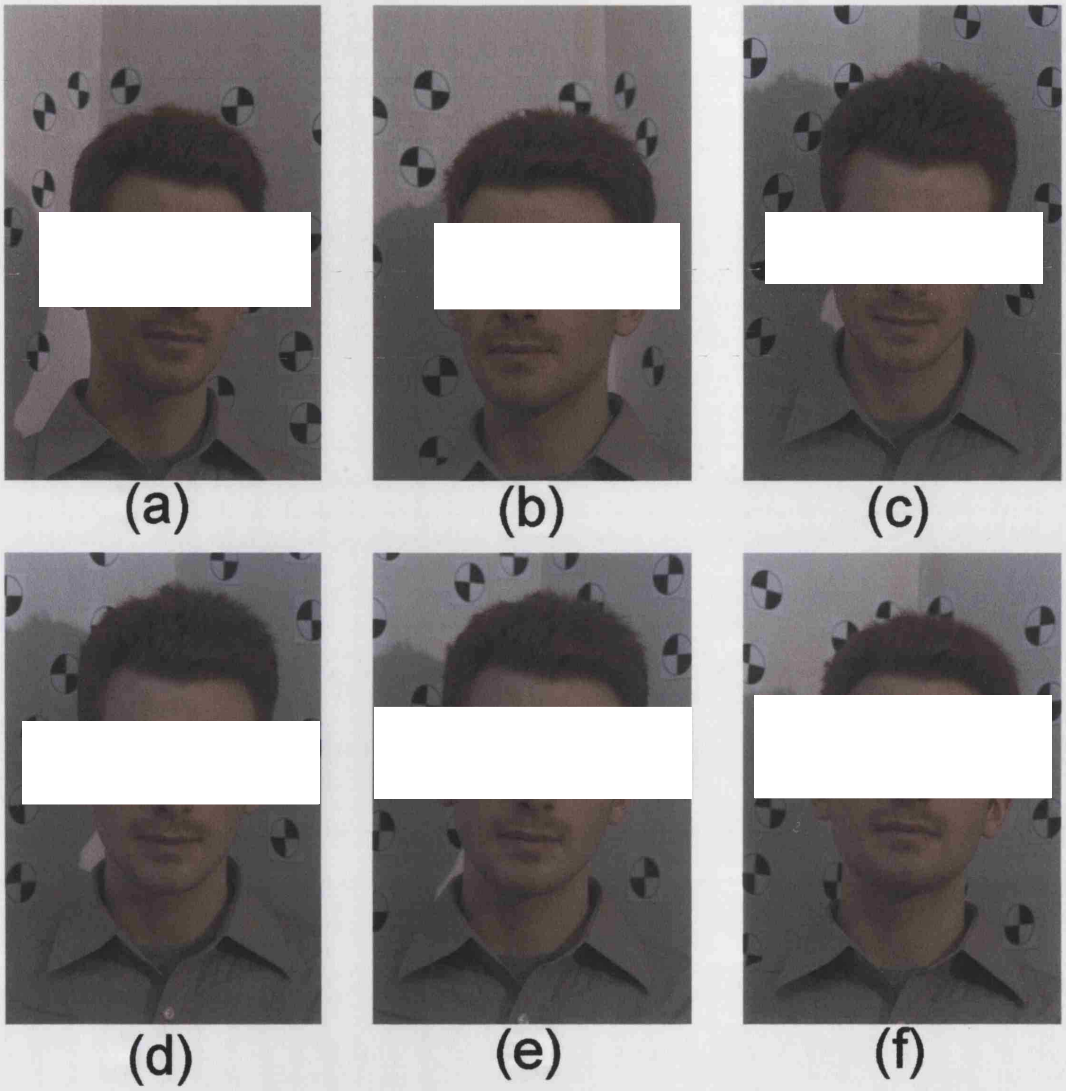


Figure 1.2. Sample views of face images.

View synthesis is closely related to techniques used in image based rendering in computer graphics [FVFH96, Len98, MB95]. Creating views of scenes from a model is of course also used in computer graphics and virtual reality. In the latter, we need to be able to render the views in real-time and objects in the foreground need to be rendered in greater detail than objects further away in the background. In order to avoid a detailed rendering of the whole scene impostors are increasingly used to allow faster generation of the view [DSSD99]. For example, an impostor can be used to represent the background of the scene using a simplified polygonal mesh where each polygon is rendered with a fixed texture. This is fast and efficient, but can cause visual artefacts in the image as the user's viewpoint is altered. Better methods of view

synthesis, especially for background objects, would be of great use. Furthermore, it is possible to combine virtual objects into real scenes. This is known as "augmented reality". An example of augmented reality is given in [SFZ99] where a logo has been added onto the side of a building and a 3D object placed on top of another building. This can be done, as in [SFZ99] by building a model of the real scene and adding virtual 3D objects. An alternative, however, would be to add the virtual objects into a small set of images and use those images in order to synthesise a variety of new views. View synthesis is also being used to create virtual or mixed virtual and real environments for immersive media, for example in videoconferencing systems [BBTS03, MZKD04, ITKO04].

View synthesis without the construction of a 3D model can be done by using warping or morphing techniques. This was done by Seitz and Dyer [SD96, SD95] in the field of computer graphics, where a new view was synthesised by linear interpolation between two existing views, a technique that could be applied, for example, to the generation of stereo displays.

As stated above, throughout this thesis we have chosen to assume that the images have been obtained using an affine camera and are generated by an affine system. It will be shown in chapter 2 that under certain viewing conditions, in particular when the variation in depth of the scene is small compared to the average depth, that the affine camera model is a good approximation to the perspective camera. Using the affine model has the advantage that the mathematics is simplified, while still leading to realistic synthesised views. Fundamentally, the simplifications of the affine system arise from the fact that, in the case of the affine camera, the 2D image co-ordinates can be written as a linear combination of the corresponding Cartesian 3D world co-ordinates in space. As a result the relationships between corresponding points in multiple affine views will also be linear.

These affine multi-view relationships have been used for the tracking of moving objects [BCZ93, BM98, FRM98, FRM95 HC94]. Given a sequence of images of a moving object, the affine relationships are used to determine the location of a set of points on the moving object in the next view. This is equivalent to synthesising the locations of a set of points in a new view and allows the camera to be moved so that it is still fixated on the object. The affine relationships are used because they are a sufficiently accurate approximation to be used in the prediction step of a tracking-

feedback loop and enable the point set to be located fast enough to allow real-time tracking of the objects of interest.

1.2 View Synthesis

The fundamental problem we study in this thesis is the generation of realistic, novel images of a scene from a set of existing images without having to construct a 3D model. We use a set of initial images that are taken with uncalibrated cameras and do not assume any measurements of the objects in the scene. Only the information that can be extracted from the images is used to synthesise the novel view.

The information that can be extracted from the images is a list of corresponding or matching points in the images. By corresponding points we mean, as in stereovision, points in two or more images that are projections of the same 3D point in the scene or on the object of interest [HZ00, TV98]. In an ideal situation we would be able to determine a dense map of such correspondences between the images and, in principle, every pixel in one image would correspond to a pixel or group of pixels in the other images, i.e., given a point in one image, we know mathematically how to map it into the other image. In practice, however, it is not possible to achieve dense correspondence over the whole image because there will be areas of the object or scene that are visible in some images but are occluded in others. Furthermore to achieve dense correspondence over image regions that are visible in all images is itself difficult, since these regions are not necessarily of the same size or shape in each image. This means that two pixels that are adjacent in the first image may correspond to two pixels in the second image that are separated by one or more pixels. In fact owing to image discretization there will almost never be any exact pixel correspondences, as they will almost always fall between pixels. Since it would be impossible to create a list of dense correspondence points we think of dense correspondence as a continuous mapping over the image regions, rather than a list of discrete points.

1.2.1 Principles of View Synthesis

In order to compute such dense mappings we make the assumption that the scene can be approximated by means of a series of planar surface patches [FL86].

This assumption allows us to define a relatively small number of sparse corresponding points, which, by analogy with image warping in computer graphics, we will call "control points". We can then warp the images between the control points using techniques similar to those used in computer graphics [Pra91]. For each surface patch it is then possible to determine a unique mapping between the images of that surface.

To see how view synthesis works, it is useful to consider the case of stereovision where, in order to obtain any information about the depths in the scene, two images are needed. It is the same with view synthesis: in order to generate a new view of a scene or object of interest, in general, we need to combine information from a minimum of two existing views. The exception to this is the case of planar objects, when it is possible to find a geometric relationship that will transform one image of the object directly into another image of it. In the general case, however, we are looking for a transformation that will map two, or more, images into a third, i.e., we are looking for a relationship between three views. It is, as noted, possible to combine more than two views in the synthesis of a novel view, but for now we will consider only the case where two views are used. Chapter 4 will explain how the algorithm can be adapted to include more views in the synthesis. The multi-view relationships between four views do not give us any more information about the structure of the scene than the three-view relationships. Therefore, we choose to perform all the experiments in this thesis by combining just two images. Henceforth, we will call the two existing views combined in the synthesis the basis views, and views being generated we will call the target views.

The process of combining the two basis views to form the target view can be broken down into three stages. Firstly the control points in the target view can be expressed as a function of the control points in the two basis views. Secondly, we use the correspondences between the control points in the three views to determine the mapping functions that provide us with dense correspondence between the target view and each basis view. Finally, we need a method for determining how the intensities in the target view should be computed from those in the two basis views.

The first stage is to determine the geometric relationships between the control points in the three views. In fact, there are actually two such relationships because in order to find the locations of the control points in the target view we need both their x and y co-ordinates. For a perspective camera the relationships between the control points in these images is known as the "trifocal tensor" [Har97a], and in the case of

the affine camera, the "affine trifocal tensor" [MC98]. Such relationships between the control points could, of course, be computed from the camera projection matrices if these were known [Har97a, TZ96], but it is advantageous, more convenient and intellectually more satisfying to infer the relationships directly from the images. In fact, study of them deepens our understanding of the relationships between three views, or as it is sometimes known, of the "viewspace". In the case of the perspective camera the relationships are non-linear [Har97a] whereas, for an affine camera they are linear. As we shall see, the latter may therefore be studied and computed by the methods of linear algebra, the matrix eigenproblem and singular value decomposition, whilst the former requires special algebraic methods [Tri96].

Fortunately, as already noted, in many real world situations it is possible to find a good approximation to the perspective camera by using an affine camera. We know that, for example, for distant objects, when the field of view is small, the affine camera can be used as a good approximation to the perspective camera. We thus choose to use the affine camera theory to find the relationships between views. The degree to which this approximation holds will be assessed in chapter 4. Using the affine camera model also has the advantage that, as noted above, the mathematics is more accessible since it keeps the relationships between the control points linear. It will also be shown that it produces more stable algorithms which are less sensitive to errors. In fact the algorithm developed by assuming an affine camera will work and produce sensible results[1] even when there are strong perspective effects in the images. This is unlike the perspective case, where algorithms may often become unstable and produce large errors when perspective effects are small though it must be said that algorithms using implicit reprojection into the image plane are less-badly behaved in this respect than most others [SA00].

It was one of our requirements that the images are taken with uncalibrated cameras. As the camera matrices are thus unknown, we cannot use them to find the view relationships directly. However, it is possible to use the general form of the affine camera matrix to determine the general form of these relationships. We will show later, in chapter 3, how the general form of the multi-view relationships can indeed be derived from the camera matrices. In fact, this will be done for both the perspective and affine cases in order to illustrate the differences between them. Given

[1] By "sensible" we mean results that look right though, of course, they will neither be correct (since the perspective relationships will not be satisfied) nor, in general, very accurate.

the general form of the multi-view relationships, it is then possible numerically to estimate the relationships between point co-ordinates in a particular triplet of images or views by using a set of corresponding control points.

This gives a method of finding the relationships between three existing views, which is useful for evaluation purposes and will be used for this purpose in chapter 4. However, this is not enough to synthesise a novel view. If we have the locations of the control points in the basis views *and* in the target view we can find the relationships between them. Alternatively if we know the control points in the basis views *and* the relationships between the three views we can find the control points in the target view. However, in the case of an unseen target view we do *not know* the relationships *or* the positions of the control points in the target view. In order to synthesise a new view we need to find *either* the positions of the control points in the unseen target view *or* the relationships between the two basis views and the novel view. Note that the latter (the multi-view relationships) like the former (the positions of the control points in the target image), change as the target view changes.

1.2.2 View Synthesis by Parameterising a Set of Sample Views

In a situation where the control points in the target view are known it is possible to estimate the relationships between the basis views and the target view [TZ97, TZ96]. An example where this situation may arise is in the use of view synthesis for video compression. Suppose we had a series of images to be transmitted, then we could choose a pair of basis views and a set of control points across all the images. The basis views could be transmitted along with the positions of a set of control points for various target views. The receiver could then use this information to synthesise each of the target views.

We have given an example of a situation where the positions of the control points in the target view are known. In general, however we may wish to synthesis a view for which the locations of the control points are unknown, i.e. a completely novel view. In this case, as stated above, we need a method that estimates either the positions of the control points in the new view or the relationships between the basis views and the novel target view.

In this thesis, we develop a method of synthesising a novel view that works by determining *either* the new relationships *or* the positions of the control points in the

novel view. To determine the new multi-view relationships we take a small number of known "sample" views, treat them as target views and, for each of them, determine a pair of view relationships relating the sample views to two *fixed* basis views. This enables us to determine a set of relationships that are valid for our sample views. We use this information to find a suitable parameterisation of the relationships in terms of the location of the target view. Having done so, we can then vary the parameters and obtain new pairs of relationships that will allow us to locate the control points in novel views for which we have no existing images. Similarly, we also use the method to parameterise the positions of the control points in terms of the location of the target view.

This transforms our problem of synthesising a novel view into that of finding a suitable parameterisation of the multi-view relationships or the positions of the control points in terms of the position of the target view. Since one of our requirements was that the cameras are uncalibrated, the parameterisation cannot rely on knowing the camera calibration, in particular the extrinsic orientation parameters [SHB99]. However it will be shown in chapter 5 that it is possible to determine a parameterisation of the control points or the multi-view relationships that does not depend on the camera parameters.

We give a method for finding a suitable parameterisation that imposes the constraint that the multi-view relationships or the positions of the control points should vary as smoothly as possible as the parameters vary. The number of parameters that can be used in the parameterisation depends on the number and nature of sample views available.

Suitable parameterisations of the multi-view relationships allow us to find the locations of the control points in a novel target view efficiently. Similarly parameterisation of the control points then allows us to determine the multi-view relationships. This was the first stage of the view synthesis procedure outlined in section 1.2.1. The next is to use the control points to determine the mapping functions that will provide the dense correspondence required in order to compute the image intensities. As mentioned earlier (section 1.2.1), in order to find the dense correspondence between the images we make the assumption that the scene is made up of planar surface patches. Even when this is not so, we will show in section 4.5.2 that this assumption still allows us to synthesise a believable image of a scene or

object that is not made up of planar surface patches, for example images of a known face.

In order to find the dense correspondence mappings, the control points are used to segment the images into corresponding regions. In the case of the affine camera we know that the mappings within each region will be affine. This means that three corresponding points are sufficient to define each mapping, and therefore we use the control points to triangulate the images. Such linear mappings are described in [Gos86] and have been used previously in [KB99] and [KB98]. In the case of the perspective camera although, as noted in section 2.6 there is still a mapping between individual images of planar surface patches, the mapping is a homography [CRZ97] and four points will be needed to define it. In this case the images are divided into quadrilateral pieces [BSG98]. In either case, the triangles or quadrilaterals should be chosen such that, where possible, they do actually lie on planar surfaces and do not include within their interiors any occluding or material boundaries in the image. In fact, it appears to be particularly important to model material boundaries as accurately as possible when there are large intensity or illuminance changes across them [Han00]. In the affine case this can be done using a constrained triangulation algorithm [Peb98].

It is also important, where possible (i.e. when there are no occlusions), to use the intensities in *both* basis views to determine the intensity in the target view. For example, only in this way can views of real objects which are not perfect Lambertian reflectors, be interpolated smoothly. By considering each triangle in the target view in turn, we are able to find a *pair* of mappings that will map that triangle onto the corresponding triangles in *each* of the basis views. For every pixel inside a triangle in the target view we then use the mappings to find a corresponding point in each basis view. When we have done this for all the triangles we have dense correspondence between the views. In general, this gives us a pair of corresponding points in the two basis views for every pixel in the target view. The corresponding points do not, of course, necessarily have to lie on a pixel, (in fact they will almost always lie between pixels) so the intensity values are interpolated in the basis views.

Once we have found the dense correspondence between the views in this way, all that remains to synthesise the novel view is decide on a method for estimating the intensities in the target image. In general, the intensities in the target image will be

some function of the intensities at the corresponding points in the two basis views [Sha92].

As noted above, we could choose one of the basis views to define the intensities in the target image and set the intensity at each control point in the target image equal to the intensity at the corresponding point in that basis view. However, if we set the target view intensities in this way, we are singling out one of the basis views and ignoring the information contained in the other view. In order to make use of the information in both basis views we need to combine the intensities from the two basis views. The simplest method would be to use an equal weighting of the two basis views. This would avoid us having to single out one of the basis views and uses all the information about the intensities.

Treating both basis views equally is a possible method of finding the new intensities and does include the information from both basis views but it may not be the most appropriate way. In particular, consider the case where the target view is more similar in appearance to one of the basis views than the other. In this case it would be appropriate to weight the intensities from that basis view more highly than the intensities from the second basis view. It is possible to get an indication of how similar the target view is to each basis view by looking at the relationships between the views [KB99]. This will be discussed in detail in section 4.4.2, where we look at how the co-efficients change as the target view is moved between the basis views. In general, if the target view is more similar to one of the basis views, the coefficients of that basis view's co-ordinates are larger than the coefficients of the second basis view's co-ordinates. If the target view is equally close to both basis views, the co-efficients of both basis views are approximately the same size. This information allows us to find an appropriate weighting of the basis view intensities based on how similar each view is to the target.

We will now outline the view synthesis method described in this thesis by giving a step by step procedure.

Outline of the Method of View Synthesis.

- Start with a set of sample images (minimum of six) and a set of control points in those images. Assume that the objects in the images are piecewise planar.

- Choose a set of variables, (E_j), to parameterise. The chosen variables must vary smoothly throughout the viewspace. Chapter 5 gives three different options for the variables, E_j:

1. The co-ordinates of the control points.
2. The co-efficients of the least squares multi-view relationships (see section 4.2).
3. The elements of the upper triangular matrix obtained from the Cholesky decomposition of the data matrix formed when finding the total least squares relationships (see sections 4.2.1 and 5.3).

- Express each of the variables, (E_j), as a function of two parameters u and v using the parameterisation method described in chapter 5.
- Choose new values for the parameters u and v. Determine the values of variables E_j. This allows us to determine both the positions of the control points and the multi-view relationships for the novel view (see chapter 5).
- Use the control points in the novel view to triangulate the image.
- Use a piecewise mapping function inside each triangle to render the intensities in the novel view as a function of the basis view intensities (see sections 4.4.2 and 3.4.2).

1.2.3 Choosing a New Basis View

In the preceding sections, we have outlined a method of view synthesis that will be explained in detail in this thesis. Our starting point is a set of uncalibrated images of an object or scene. We begin by choosing a pair of basis views and a set of corresponding control points in the images and use this information, along with the affine imaging assumptions and the assumption that the scene is piecewise planar, to synthesise a novel view.

The view synthesis method described here could, in theory, generate a novel view at an arbitrary distance from the chosen basis views. However, in practice, as we would expect, the procedure works better when the novel view is close to the basis views. As we move away from the basis views the synthesised view will become progressively worse, for example less realistic looking, or a less accurate approximation to an existing view. We would thus like to be able to get an estimate of

how far we can extrapolate from the basis views before the procedure breaks down. We will show in chapter 6 that it may be possible, in some cases, to use the structure contained within the method of determining the multi-view relationships, to determine a limit of extrapolation. This allows us to determine at what point it is necessary to choose a new basis view or pair of basis views.

1.2.4 Limitations

The starting point for the method of view synthesis described in section 1.1.2 is a set of corresponding control points. Determining this set automatically is a challenging task, which is outside the scope of this research. Although there are methods that solve the correspondence problem [DSST00, Pil97], throughout this thesis the control points used in the experimentation are hand picked. By hand picking the control points we avoid large errors and bogus correspondences. This allows us to distinguish between the limitations in the view synthesis procedure from errors introduced by poor correspondence. Similarly, triangulation of the control points was performed manually.

1.3 Applications

There are many situations where being able to synthesise a new view, without the construction of a 3D model, would be useful. Although this thesis is concerned with the view synthesis method and not a specific application, here we discuss briefly some possible uses for the procedure.

1.3.1 Tourism Industry

First, there are applications where the procedure would enable the cost involved in building a 3D model to be avoided. Synthesising new views from existing images requires less computation than first generating an accurate 3D model. This allows faster generation of the novel view while still producing realistic images. An example where this could be particularly useful is in the tourism industry. In choosing a resort most people would find a photograph more useful than a simple 3D model. A

simple 3D model that allows the viewer to rotate around a fixed viewpoint produces realistic images but gives no impression of space. In order to obtain an idea of space the viewer needs to be able to alter the viewpoint [Bux01]. While the photograph only gives a limited impression of the 3D structure, the level of detail within a photograph is much more valuable to the viewer. In contrast an extremely detailed 3D model could approach the same quality level and give true 3D structure but it would probably be prohibitively expensive to produce. View synthesis from existing images can combine the advantages of the photograph with the sense of 3D structure. It is possible to synthesise views from adjoining positions to produce a sequence similar to a video clip, but without the fixed viewpoint trajectory. The viewer can choose in which direction they wish to move the camera.

1.3.2 Sales Applications

Other possible applications of this kind of view synthesis, which take advantage of the high realism and the impression of 3D structure that it generates, include retailers such as estate agents and those selling furniture online. A set of photographs of a house could be used, in a similar way to those of a holiday resort, to allow the user to get a feel for each of the rooms in the house. Again the important point is that view synthesis can create a better impression than from a limited number of photographs and gives a sense of realism that could not be captured in a simple 3D model of the house. In particular, when buying furniture people are interested in the detailed texture of the material as well as the size and shape and using view synthesis would allow the customer to get an impression of both. It would, for example, be possible to walk around the furniture and view it from a variety of angles, as well as being able to see details in the finish of the materials.

1.3.3 Historic Applications

Alternatively, view synthesis could be used in learning about our general cultural heritage to generate images of historical sites or museum exhibits from around the world [Han99, HB00a, HB00b]. Pictures of historical buildings or places are not able to give an impression of the overall structure. View synthesis could be used to generate a series of images and let the user walk round the site and view the

scene from different angles [PKVV98]. This allows the user to get a more realistic impression of the place of interest without having to travel great distances to visit the site.

1.3.4 Virtual Reality Applications

We mentioned earlier, in section 1.1.1, the use of impostors when synthesising scenes in virtual reality environments. It was pointed out that using impostors to represent the background of a scene can lead to visual artefacts in the image as the viewpoint is altered. It should be possible to combine the view synthesis methods with the use of the impostors to produce smooth changes in the images. Impostors could be used to synthesise the background from a small set of viewpoints. View synthesis could then be used to synthesise the background at the intermediate viewing positions and produce smooth changes in the background. The view synthesis method would only be used to produce the background of the scene, objects in the foreground would still be rendered from the 3D model. Applications where the view synthesis could be used in virtual environments in this way include flight simulators, architectural visualisation and computer games.

1.3.5 Videoconferencing Applications

View synthesis can be used to compress data in situations where it is not possible to transmit a complete set of full images. An example where this might be useful is communication by videophones [KB99]. In this situation the picture is usually a face image. Changes to this image will be due to changes in the facial expression. This leads to a limited number of changes being made to the images. In order to avoid transmitting a complete image every time the picture was updated, it would be possible to use a small set of images and use view synthesis to update the image. As described in [KB99, KB98] a small set of images of the face could be stored along with the locations of a set of control points in each image. In order to update the image at the receiver it is possible to send the locations of the control points in the new image. The new locations of a set of control points could be used to synthesise the image from the set of stored images.

The view synthesis procedure could also be used to create environments for immersive videoconferencing systems [BBTS03, ITKO04, and MZKD04]. The users of an immersive system are immersed in a shared environment with a strong sense of social and physical presence. For example, the system may create an environment where the users all seated around a shared table. The environment that is shared by the users can be a virtual space or a mixed reality (combined virtual and real) space [ITKO04]. Each of the participants head movements and gestures are continuously estimated by a head tracker and rendered into the virtual environment. The individual views of the conference scene are then synthesised at each of the user's terminals from the appropriate viewpoint.

1.3.6 Film Industry Applications

If we have a sequence of images of an object taken with several cameras placed at intervals around the object it would be possible to use view synthesis to interpolate between the images. The images could then be combined to produce a video clip that rotates around the object. This could be used to create special effects for film and television. Special effects have been created in a similar way, by having the cameras close together and using morphing techniques [SD96] to interpolate between the images. Interpolating the views using view synthesis has the advantage that the cameras do not need to be placed as close together as it produces more accurate results than a simple warping technique.

1.3.7 Object Recognition

Another area where it is possible to make use of the view synthesis techniques is in object recognition [UB91, Ull96] where we wish to recognise an image of an object by using a direct comparison with an image of a known object. Suppose we have an image of an unknown object and we would like to compare this with a database of stored images to try and determine what it is. In order to compare the unknown image with an existing one it is necessary to make sure that the object appears in the same orientation in both images. It is possible to do this by using two existing images to synthesise a novel view from the same orientation as the view we are trying to recognise. Key features in the images can be used as the control points.

These control points can then be used to combine the two known views to synthesise a new view of the known object in the same orientation as the object in the unknown image. It is then possible to make a direct comparison of the two images. Examples where view synthesis could be useful for recognition include, for example, robot navigation and criminal investigations. If a robot were trying to navigate its way around it would be useful to recognise objects in its path and know whether it had seen them before. Recognition techniques are used in criminal investigations for comparing security camera footage of the criminal with photos of a suspect. It is unlikely that the two images of the suspect will be in the same orientation and this is where the view synthesis method could be of assistance.

1.4 Outline of the Thesis

We began this chapter by giving an outline of the view synthesis procedure. Our starting assumptions were that all the images were taken with an affine camera. In chapter 2 we thus give a description of the perspective and affine camera models, and derive the affine model as a special case of the perspective camera with its centre on the plane at infinity. The form of the affine camera is an important point of this chapter, as this allows us to write the co-ordinates of a point in the image as a linear combination of the 3D world co-ordinates of the corresponding point in space. These linear equations will be used in section 3.3.1 to derive the affine multi-view relationships. Of particular interest are the relationships between three affine views. These can be represented using the affine trifocal tensor, which will be discussed in section 3.3.2. The three-view relationship in the perspective case, known as the trifocal tensor, is given in section 3.2.1 so that we are able to make a comparison between the affine and perspective cases.

In chapter 4 we give the details of the first stage of the view synthesis method, that is, how we estimate the multi-view relationships. In this chapter, we introduce the total least squares techniques that should be used to estimate the multi- view relationships from a set of control points in the images. We then evaluate the total least squares technique by comparing with the ordinary least squares solution using a set of sample images.

In order to find the relationships between two known basis views and an unknown target we need to parameterise either the relationships between known

views or the control points. A method for parameterising the relationships or the control points is given in chapter 5. We also evaluate and compare the synthesised images that are obtained when the method is used to parameterise the relationships and the control points. Tests are carried out using both synthetically generated data and real images.

In chapter 6 we show that in some cases it may be possible to use the structure within the multi-view relationships to get an estimate of a limit of extrapolation. This may help us to determine at what point the view synthesis procedure breaks down and indicates when another basis view or pair of basis views should be chosen in order to still be able to synthesise a perceptually convincing image. We finish, in chapter 7, with the conclusions and suggestions for further work.

1.5 Aims and Contribution

The main aim of this thesis is to develop a robust method of view synthesis. To do so, we use two assumptions. The first is that all the images have been obtained under affine viewing conditions and that those to be synthesised will also satisfy the affine imaging conditions. The second assumption is that the scene is piecewise planar. The objective is to be able to produce, from a limited number of initial views, believable novel views using these assumptions. The subsidiary aim of this thesis is to be able to assess how good the novel view synthesised will be. In other words, we would like to determine whether the view we are trying to generate will lead to a plausible image or whether a new pair of basis views should be chosen.

In this thesis we make three main contributions. The first is the way in which we estimate the relationships between existing views. We form a new set of affine multi-view relationships by treating all the views on the same footing. By treating all the views "the same" we mean that we seek a pair of relationships that involve all co-ordinates in the target view and both basis views in a similar way, i.e., we wish the relationships to be symmetrical in the co-ordinates of each of the views. Previous multi-view relationships have singled out the two target view co-ordinates, and obtained two explicit relationships, one equation for the x target view co-ordinate and one equation for the y co-ordinate.

The reason for forming relationships that treat both the basis views and the target view on an equal footing arises from the fact that, for existing images, all points

in all views are equally likely to be affected by the same measurement errors. There will inevitably be some measurement errors involved in locating the control points. These errors will occur whether the control points are located manually or automatically. In order to minimise the errors, we use the appropriate total least squares formula [VV91], as opposed to the ordinary least squares techniques [GV96, LH95] that have been used previously. By considering where the errors occur and minimising these errors appropriately we are able to obtain more accurate relationships between the views as shown in section 4.3.

The second contribution that we make is the method of synthesising novel views by finding a suitable parameterisation of a set of sample views. In chapter 5 we show how the method can be used to parameterise three different sets of variables, two different pairs of multi-view relationships and the positions of the control points. We compare the results of the view synthesis method when it is used in the three different ways in section 5.6.

Finally, in chapter 6 we show that in certain cases, in particular when the images are of a symmetrical object, there may exist a point where the multi-view relationships breakdown and can no longer be used to solve for the locations of the control points in the target view. We show that it may be possible to use the structure within the total least squares problem to predict the point at which the relationships breakdown.

Chapter 2

Projective Geometry, Camera Models and Two-View Relationships

The starting point for many computer vision problems is the camera model. We begin this chapter by introducing the perspective camera model, which is widely used in computer vision [HZ00, TV98]. By using homogeneous co-ordinates, the camera model can be represented as a matrix. Homogeneous co-ordinates are introduced in section 2.1.2. We use the matrix form of the camera models to derive the form of the multi-view relationships. Section 2.2 discusses the geometry of two perspective views and we introduce the idea of epipolar geometry. The epipolar geometry allows us to form a relationship between a point in one image and a line (the epipolar line) in the second image on which the corresponding point will lie. This relationship between a point in one image and a line in the other image can be expressed, for calibrated cameras, in the form of the essential matrix, or in the case of uncalibrated cameras as the fundamental matrix. The essential matrix and fundamental matrix are derived from the projection model in sections 2.2.2 and 2.2.3 respectively.

The view synthesis method developed in this thesis is based on the affine camera model as this has the advantage of producing more stable algorithms. The affine camera model is presented in section 2.3. The general form of the affine model includes the orthographic, the weak perspective and the para-perspective cameras. The properties of the affine camera are discussed at the end of the section. Section 2.4 discusses the geometry of two affine views and it is shown how the epipolar geometry is simplified in the affine case.

In the first chapter we made the assumption that the scene is locally planar. There are perspective mappings that exist between planar surfaces in two images. These mappings are known as homographies and are described in section 2.5. We also discuss what happens to these mappings in the affine case as they will be used later in

order to obtain the dense correspondence needed to reconstruct the intensities when synthesising a novel view.

2.1 The Perspective Camera

We start in section 2.2.1 by describing perspective projections [SK98]. These are projections of points in space onto an image plane; i.e. they transform a 3D point into a 2D point.

2.1.1 Perspective Projections

We start by describing a simple pinhole camera [HZ00, TV98] and do not allow for any lens distortion. This gives us the basic perspective projection equations, which we will use later when we build the complete camera model. Consider a camera with its centre at a point O_c and co-ordinate axes X_c, Y_c and Z_c (figure 2.1). The camera centre, O_c, is also known as the centre of projection. We consider a point in space P_c with co-ordinates (X_c, Y_c, Z_c) that is being imaged by this camera onto the image plane at point p_c. The Z_c axis of the camera is known as the optical axis and is perpendicular to the image plane. The distance from the camera centre to the image plane is the focal length of the camera denoted by f.

The point in the image plane, p_c, is the point where the line joining the optical centre of the camera O_c and the point in space P_c intersects with the image plane. The point p_c has co-ordinates (x_c, y_c, z_c) in the camera co-ordinate system, where z_c takes the constant value f for every point projected onto the image plane. Since this value remains constant we are only interested in the first two co-ordinates of p_c and therefore points in the image plane can be described as 2D points by dropping the last co-ordinate, i.e. p_c can be written as (x_c, y_c).

We know that the points p_c and P_c both lie on the same line from the optical centre of the camera. This means that it is possible to write one as some multiple of the other:

$$(x_c, y_c, f) = \lambda(X_c, Y_c, Z_c) \quad . \qquad (2.1.1)$$

We can see from equation (2.1.1) that $\lambda = f/Z_c$, and we can write the two co-ordinates of p_c that we are interested in as a function of the co-ordinates of P_c as:

$$x_c = f\frac{X_c}{Z_c} \quad , \qquad \text{and} \qquad y_c = f\frac{Y_c}{Z_c} \quad . \tag{2.1.2}$$

These equations are the basic perspective projection equations. It can be seen in figure 2.1 that it is possible to write down the equations for x_c and y_c by using the fact that the ratios of the sides of the similar triangles will be the same.

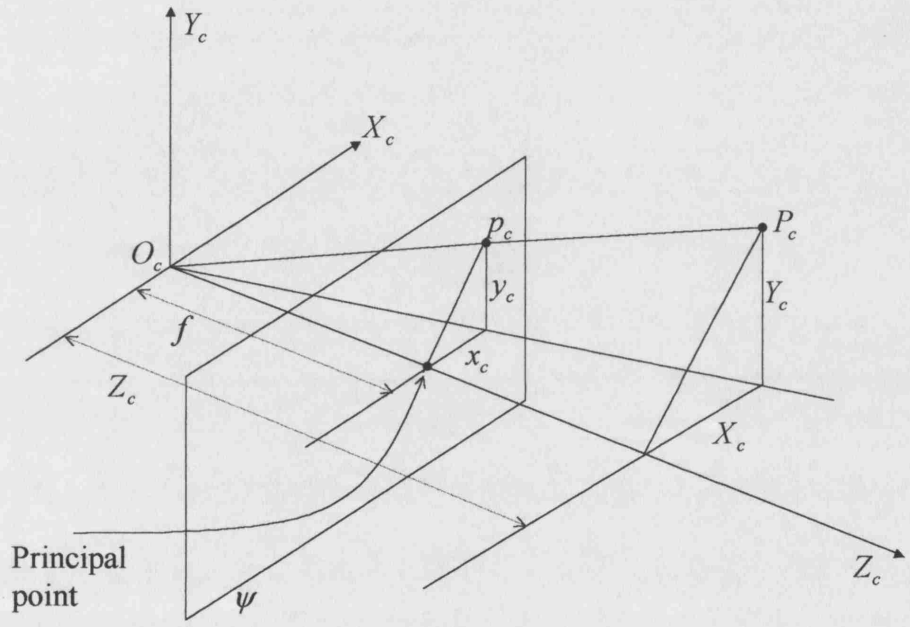


Figure 2.1. Perspective projection of a point in space, P_c onto image point p_c.

2.1.2 Homogeneous Co-ordinates

By using homogeneous co-ordinates, the pair of equations 2.1.2 can be written as a single matrix transformation. A 3D point in space (X,Y,Z) can be written in homogeneous co-ordinates as $(WX,WY,WZ,W)^{\mathrm{T}}$ where W is a non-zero weighting or scaling factor. Similarly in 2D space a point (x,y) can be represented using homogeneous co-ordinates as $(wx,wy,w)^{\mathrm{T}}$.

If we use this representation of points it is possible to write the pair of non-linear perspective projection equations (2.1.2) as a single 3×4 transformation matrix:

$$p_c = \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} = \frac{1}{Z_c}\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \frac{1}{Z_c}\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} P_c \quad . \qquad (2.1.3)$$

However, in homogeneous co-ordinates two representations of a point are equivalent if one is a multiple of the other:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \cong \begin{pmatrix} wX \\ wY \\ wZ \\ w \end{pmatrix} \text{ for } w \neq 0 \quad , \qquad (2.1.4)$$

where "$\cong$" is used to represent equality up to a scaling factor. By using this equivalence we can drop the scaling factor $1/Z_c$ in equation (2.1.3) which then becomes:

$$p_c = \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} \cong \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} P_c \quad . \qquad (2.1.5)$$

The projection model given in (2.1.5) is known as the ideal pin-hole camera model. In what follows we will allow 2D points to be expressed using either the 2D co-ordinate representation (x, y) or the homogeneous representation $(x, y, 1)^T$. For example the 2D point p_c can be written as (x_c, y_c) or as $(x_c, y_c, 1)^T$. Whether a point is represented using homogenous co-ordinates or not will be stated or will be obvious from the context in which it is used. Similarly 3D points may be written as either (X, Y, Z) or $(X, Y, Z, 1)^T$.

2.1.3 The Perspective Camera Matrix

The perspective projection equations in (2.1.2) and the matrix in (2.1.5) describe the transformation of a point in space P_c onto a point p_c in the image plane. Both the points P_c and p_c are described using a co-ordinate system defined by the camera.

However, the choice of co-ordinate system attached to the camera centre was very special. In order to build a more general camera model we need to allow for two more transformations, changes of co-ordinate system in the image plane and of the point in space.

First we consider the co-ordinate system in the image plane. The optical axis of the camera passes through the image plane at the principal point [HZ00] and has co-ordinates $(0,0)$ when described using the above camera co-ordinate system. Usually, however, the image point will be defined using image co-ordinates, otherwise known as pixels. The origin in image co-ordinates will not necessarily be at the principal point and the pixel size may be different from the camera co-ordinate units, so we also need to allow for a translation and scaling of the image points. Although it is common to assume that the image axes are orthogonal, this may not necessarily be the case. To allow for skewed axes we also include a shear term in the transformation from camera co-ordinates into pixels.

We know that our projected point p_c is at a position (x_c, y_c) in camera co-ordinates. When this point is referenced in image co-ordinates we will call it p and denote its location as (x, y). The translation, scaling and shear of the axes used to transform p_c to p can be written as a pair of equations:

$$\begin{aligned} x &= o_x + s_x x_c + \alpha y_c \\ y &= o_y + s_y y_c \end{aligned} \qquad (2.1.6)$$

(o_x, o_y) is the pixel location of the principal point, s_x and s_y are the scaling factors from the camera co-ordinates to the image co-ordinates, and α is the shear term.

If we return to our homogeneous co-ordinates notation we can write this pair of equations as a 3×3 matrix as:

$$p = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & \alpha & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} \qquad (2.1.7)$$

It is quite common to assume that the scaling factors s_x and s_y are equal, as for example when the camera used has square pixels.

If we multiply the transformation matrix in (2.1.7) by the perspective projection matrix in (2.1.5) we obtain a 3×4 matrix. The resulting matrix transforms

a point in space defined in camera co-ordinates to the pixel location of the point in the image:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} s_x & \alpha & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$
$$= \begin{pmatrix} fs_x & f\alpha & o_x & 0 \\ 0 & fs_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \qquad (2.1.8)$$

The 3×4 matrix in (2.1.8) above is known as the matrix of intrinsic camera parameters, M_{int} [MZ92]. For any given camera the intrinsic parameters in general remain constant as the position of the camera is altered. However if the camera has a zoom then this means the intrinsic parameters may vary [dAHR98].

Equation (2.1.8) generalises the ideal pin-hole camera model (2.1.5) by taking account of the intrinsic parameters of the camera. However, in general, as well as the intrinsic parameters we have extrinsic parameters which relate to the position of the camera in space, rather than to the projection process. Unlike the intrinsic parameters which change only when the internal set-up of the camera is changed, the extrinsic parameters change as the position of the camera in space is altered and are therefore defined relative to the choice of some external frame of reference, in practice supplied by a convenient calibration object [TV98].

Recall from section 2.1.1 our point in space P_c. This point was defined in the camera co-ordinate system. Now suppose that we move the camera and wish to image the point in space from the new camera location. In order to do this we need to redefine our point in the co-ordinate system defined by the new camera position. We do this by setting up a co-ordinate system in space that we will refer to as a world co-ordinate system. Our point in space will be written as P in world co-ordinates and its position denoted by (X,Y,Z). The world co-ordinate system will remain constant for every camera position. We can then transform the world co-ordinate frame into the camera co-ordinate frame by using a translation, such that the origins of the two systems coincide, and a 3D rotation to ensure that each of their axes lies in the same direction in both systems. This transformation is illustrated in figure 2.2 and can be written as:

$$P_c = R(P - T) \quad , \tag{2.1.9}$$

where R is a 3D rotation and T is a translation vector that describes the position of the camera origin relative to the origin of the world co-ordinate system. Again it is possible to re-write equation (2.1.9) as a matrix by using the homogeneous co-ordinate representation of the points:

$$P_c = \begin{pmatrix} X_c \\ Y_c \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} R & -RT \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} R & -RT \\ 0 & 1 \end{pmatrix} P \quad . \tag{2.1.10}$$

This gives us our 4×4 matrix of extrinsic camera parameters, M_{ext}.

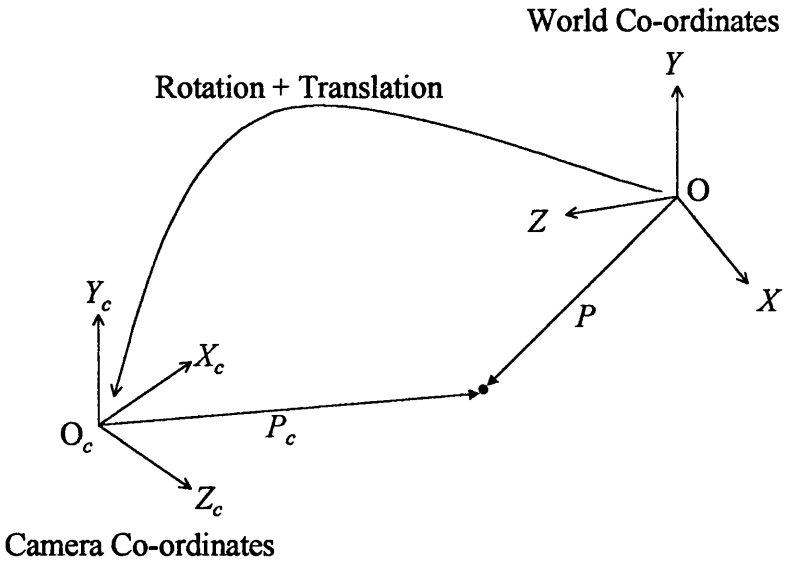


Figure 2.2. Change of 3D co-ordinate system.

In order to obtain the perspective camera matrix, M, we multiply the intrinsic and extrinsic parameter matrices,

$$\begin{aligned} M &= M_{\text{int}} M_{ext} \\ &= \begin{pmatrix} fs_x & f\alpha & o_x & 0 \\ 0 & fs_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & -RT \\ 0 & 1 \end{pmatrix} \end{aligned} \quad , \tag{2.1.11}$$

which gives the general form for the perspective projection matrix, M, that transforms world points, P, into image points, p.

$$p \cong MP$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad . \qquad (2.1.12)$$

The projection matrix, M, in (2.1.12) has 12 elements but the overall scale of the matrix is unimportant which means that M has 11 degrees of freedom. If we look at (2.1.11) it look like there are 12 degrees of freedom, (3 in the rotation matrix R, 3 in the translation T and 1 for each of f, s_x, s_y, α, o_x and o_y). However if we look at the intrinsic parameter matrix we can see that the focal length only appears when it is multiplied by one of the scaling terms or the shear term, therefore this matrix only has five degrees of freedom which are the terms fs_x, fs_y, $f\alpha$, o_x and o_y.

2.2 The Geometry of Two Perspective Views

Now that we have the general perspective camera model we can explore the geometry of multiple views. Two-view geometry is used in photogrammetry [Atk96] to determine the 3D co-ordinates of an object. In photogrammetry the cameras are usually very carefully calibrated [Bey92, Tsa86] and the geometry is known. When the cameras are uncalibrated 3D geometry may no longer be unambiguously recovered but it is possible, for example, using two views to determine the structure of the scene up to a projective ambiguity [Fau92, HZ00]. In what follows we describe the relationships between the two calibrated views and then extend this to the uncalibrated case.

2.2.1 Epipolar geometry

We begin by introducing the notation we shall use to describe the epipolar geometry. The set-up is shown in figure 2.3. In this figure point P in space is being imaged by two cameras with centres at O_c and O_c'. We will distinguish between the two cameras by referring to them as left "L" and right "R". The respective image planes are ψ and ψ'. Point P will be referred to as P_c and P_c' when referred to the left and right camera co-ordinate frames respectively. The image points will be written as p_c and

p'_c when referred to the camera frames and as p and p' in the respective left and right image co-ordinates.

The intrinsic parameter matrices of the left and right cameras will be denoted by M_{int} and M'_{int} respectively. Similarly the extrinsic parameter matrices of the left and right cameras will be denoted M_{ext} and M'_{ext}.

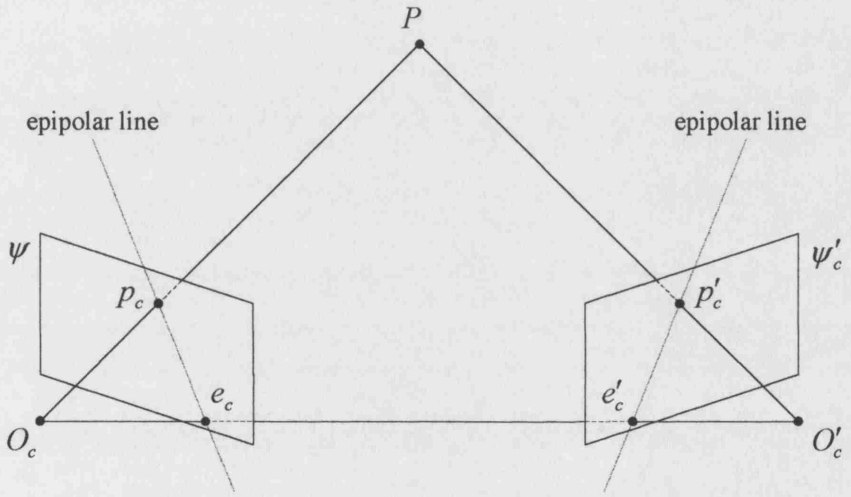


Figure 2.3. Epipolar Geometry.

The line joining the two optical centres O_c and O'_c (known as the baseline) passes through the image planes ψ and ψ' at points e_c and e'_c in camera co-ordinates (or points e and e' in pixel notation). These points are known as the epipoles. As can be seen from figure 2.3, e_c is the projection of camera centre O'_c onto image plane ψ and e'_c is the projection of O_c onto image plane ψ'.

The epipolar plane for the point P is defined to be the plane passing through P, O_c and O'_c. The epipolar lines are the lines in which the epipolar plane intersects each image plane. We denote these lines by l and l' respectively. In the left image

plane the epipolar line, l, passes through points e_c and p_c. Similarly in the right image plane the epipolar line, l', passes through e'_c and p'_c.

It thus follows that each point in the left image, p_c, defines an epipolar line, l', in the right image which must contain the corresponding point, p'_c. This is known as the epipolar constraint [Fau93] in stereovision and is very important, for example, in reducing the complexity of the matching problem of finding corresponding pairs of points.

2.2.2 The Essential Matrix

The essential matrix, first introduced in computer vision by Longuet-Higgins [L-H81], provides an explicit relationship between the co-ordinates of corresponding points in two images when the points are described in the appropriate camera co-ordinate system [HF89, L-H81, TV98]. By using the camera co-ordinate systems we are assuming that the intrinsic parameters M_{int} and M'_{int} are known and only $M_{\text{ext}}^{-1}M'_{\text{ext}}$ or $M'^{-1}_{\text{ext}}M_{\text{ext}}$ are required to determine the epipolar lines. The more general fundamental matrix will be discussed in the next section.

The points P_c and P'_c both refer to the same point in space, P, but are defined in their own camera co-ordinate frames. It is possible to write down a transformation between the two camera co-ordinate systems; i.e., a transformation from P_c to P'_c. This transformation has the same form as equation (2.1.9), since it consists of a translation of the origin from one camera's co-ordinate system to the other and a rotation to align their axes. Thus,

$$P'_c = R(P_c - T) \quad . \tag{2.2.1}$$

For the moment, we drop the homogeneous co-ordinates and use P_c and P'_c to stand for ordinary vectors in 3D space. R is thus a 3×3 rotation matrix and T is the (3-vector) translation between the two camera centres, $(O'_c - O_c)$.

We know that the epipolar plane for point P passes through the points P_c, O_c and O'_c, and that $T = O'_c - O_c$. We can therefore write the equation for the epipolar plane as a coplanarity condition on the three vectors P_c, T and $(P_c - T)$:

$$(P_c - T)^T T \times P_c = 0 \quad . \tag{2.2.2}$$

Equation (2.2.1) above can be rearranged as $(P_c - T) = R^T P_c'$ and substituted into (2.2.2) above to obtain:

$$(R^T P_c')^T T \times P_c = P_c'^T RT \times P_c = 0 \quad . \tag{2.2.3}$$

By using the fact that a vector product can be written as multiplication by an antisymmetric rank deficient matrix [L-H81] we can replace the vector product in (2.2.3) with S. Equation (2.2.3) then becomes:

$$P_c'^T RSP_c = 0 \quad , \tag{2.2.4}$$

where $S = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix}$. The matrix RS is known as the essential matrix and is denoted by E.

We know from equation (2.1.2) that image points p_c and p_c' can be written as:

$$\begin{aligned} p_c &= \frac{f}{Z_c} P_c \\ p_c' &= \frac{f}{Z_c'} P_c' \end{aligned} \quad . \tag{2.2.5}$$

The third component of each of (2.2.5) are not very informative in that they amount only to the identity $f = f$, but if we substitute for P_c and P_c' in equation (2.2.4) in terms of p_c and p_c', and divide by $Z_c Z_c' / f^2$ we obtain a relationship between the image points [L-H81]:

$$p_c'^T E p_c = 0 \quad . \tag{2.2.6}$$

The relationship in (2.2.6) is between a pair of corresponding points in the images. Since the epipolar lines satisfy a similar coplanarity condition, it is also possible to write down a relationship between a point in one image, p_c, and its epipolar line, l_c', in the second image which may be expressed as:

$$E p_c = l_c' \quad . \tag{2.2.7}$$

l_c' is the epipolar line in the right image passing through the epipole e_c' and the point corresponding to p_c in the right image, p_c'. As noted previously in section 2.2.1, the relationship between a point and its epipolar line can be used when searching for

corresponding points. Given the essential matrix and a point in the left image it is possible to find the epipolar line in the right image on which the corresponding point will lie. This reduces the search for correspondence to a 1D problem [TV98]. Similarly if a point in the right image is known and we are searching for a matching point in the left image we can use the essential matrix to provide the corresponding left epipolar line:

$$p_c'^T E = l_c^T \quad , \tag{2.2.8}$$

where l_c is the line passing through the point p_c and the left epipole, e_c.

2.2.3 The Fundamental Matrix

In order to compute the essential matrix from point correspondences in the images we need to be able to find p_c and p_c' from their pixel locations p and p'. To do this we need to know the intrinsic camera parameters. The fundamental matrix differs from the essential matrix in that the intrinsic parameters as well as the extrinsic parameters [Fau92, Har92] are unknown. Since the fundamental matrix provides us with a relationship between p and p', it can be estimated from a set of corresponding points, without the need to calibrate the cameras.

We will now derive the fundamental matrix [HZ00, TV98]. We assume the set up illustrated in figure 2.3. Again we drop the homogeneous co-ordinates notation for our 3D points P_c and P_c', however we keep our homogeneous notation for the 2D points i.e., $p = (x, y, 1)^T$ and $p' = (x', y', 1)^T$. This means that our intrinsic camera matrices become 3×3 matrices. These can be obtained by removing the last column of zeros in the matrix in (2.1.8). We know that our intrinsic camera matrices transform P_c and P_c' to p and p',

$$\begin{aligned} p &\cong M_{\text{int}} P_c \\ p' &\cong M'_{\text{int}} P_c' \end{aligned} \quad . \tag{2.2.9}$$

We can re-arrange equations (2.2.9) to get expressions for P_c and P_c', and then substitute these into the equation for the essential matrix (2.2.4), to yield:

$$p'^T \left(M'^{-1}_{\text{int}}\right)^T E M^{-1}_{\text{int}} p = 0 \text{, i.e. } \quad p'^T F p = 0 \quad . \tag{2.2.10}$$

The matrix F in this relationship between p and p' is known as the fundamental matrix. It can be written as a combination of the two intrinsic parameter matrices and the essential matrix:

$$F = \left(M'^{-1}_{\text{int}}\right)^T E M^{-1}_{\text{int}} \quad . \tag{2.2.11}$$

As with the essential matrix, given a point in one image the fundamental matrix constrains the corresponding point in the other image to lie on the epipolar line. It is possible to estimate the fundamental matrix from a set of correspondences in the two images. A method of estimating the fundamental matrix can be found in section 3.4.1.

The fundamental matrix can be used in the matching problem of uncalibrated cameras to reduce the search for corresponding points to a 1D problem. Given a point in the left image, p, and the fundamental matrix, F, we are able to determine the epipolar line, l' in the right image which contains the corresponding point, p':

$$Fp = l' \quad . \tag{2.2.12}$$

Similarly, we can write an equation between the point p' in the right image and the epipolar line, l, in the left image:

$$p'^T F = l^T \quad . \tag{2.2.13}$$

If we replace the point p in (2.2.12) with the left epipole e, we find that the right hand side is equal to zero:

$$Fe = 0 \quad , \tag{2.2.14}$$

and that e is thus the right null space of F. Similarly, replacing p' in (2.2.13) with the right epipole, e' we find that the right hand side is equal to zero:

$$e'^T F = 0 \quad . \tag{2.2.15}$$

The right epipole e' is thus the left null space of F [Fau92].

2.3 The Affine Camera

Under certain viewing conditions it is appropriate to use approximations to the full perspective camera, in particular when the variation in depth of the scene is small compared to the average depth of the scene. The view synthesis method described in this thesis has been developed using the affine camera theory discussed here. An affine camera can be thought of as a perspective camera with its optical centre on the plane at infinity [HZ00, SZB95]. In section 2.3.1 we describe the orthographic

camera, which is the simplest form of an affine camera. We then move on to the more general scaled orthographic and weak perspective cameras [SZB95, Zis92] in section 2.3.2. The orthographic and weak-perspective are specific examples of the affine camera which will be discussed. The general form of the affine camera also includes the para-perspective or oblique camera model. In the case of the orthographic and weak perspective cameras the lines of projection are perpendicular to the image plane. The general case where the lines of projection are not perpendicular to the image plane is known as the para-perspective or oblique projection. The para-perspective camera is not discussed here but details can be found in [SZB95] and [Alo90]. The general form of the affine camera is summarised and its properties are discussed in section 2.3.3.

2.3.1 The Orthographic Camera

Consider a perspective camera with its optical centre on the plane at infinity. Since the optical centre is an infinite distance from the optical plane and from the object points in space, all lines of projection become parallel and the orthographic projection is sometimes called parallel projection. Under orthographic projection the depth of the scene is ignored completely.

If we consider our perspective projection equations (2.1.2),

$$x_c = f\frac{X_c}{Z_c} \quad , \qquad \text{and} \qquad y_c = f\frac{Y_c}{Z_c} \quad , \tag{2.3.1}$$

and let both the focal length, f, and the depth, Z_c, both tend to infinity, then f/Z_c tends to one. In the orthographic case our projection equations thus become simply:

$$\begin{aligned} x_c &= X_c \\ y_c &= Y_c \end{aligned} \tag{2.3.2}$$

Two points in space project to the same image point if they only differ in the Z_c co-ordinate. In an orthographic camera there is no scaling between the camera co-ordinates and pixels but we do allow for a translation of the origin so our pixel co-ordinates can be expressed as:

$$\begin{aligned} x &= x_c + o_x = X + o_x \\ y &= y_c + o_y = Y + o_y \end{aligned} \tag{2.2.3}$$

Thus our 3×4 intrinsic parameter matrix can now be written as:

$$M_{\text{int},or}=\begin{pmatrix}1&0&0&o_x\\0&1&0&o_y\\0&0&0&1\end{pmatrix} \quad . \tag{2.3.4}$$

The extrinsic parameter matrix, M_{ext} (2.1.10) remains the same. Multiplying together $M_{\text{int},or}$ and M_{ext} gives us our orthographic camera matrix,

$$\begin{aligned}M_{or}&=\begin{pmatrix}1&0&0&o_x\\0&1&0&o_y\\0&0&0&1\end{pmatrix}\begin{pmatrix}R&-RT\\0&1\end{pmatrix}\\&=\begin{pmatrix}r_{11}&r_{12}&r_{13}&-(RT)_1+o_x\\r_{21}&r_{22}&r_{23}&-(RT)_2+o_y\\0&0&0&1\end{pmatrix}\end{aligned}\quad , \tag{2.3.5}$$

where r_{ij} is the element from the i^{th} row and j^{th} column of the rotation matrix R and $(RT)_k$ is the k^{th} entry of the vector RT.

2.3.2 Scaled Orthographic and Weak Perspective Cameras

The orthographic camera does not allow any scaling between the camera co-ordinates and pixels. If we allow such a scaling between the two co-ordinate systems the result is a scaled orthographic or weak perspective camera [Sha95].

If we have a scene where the variation in depth, $\Delta Z_c = Z_c - \overline{Z}_c$, is small compared to the average depth, $\overline{Z}_c$, then a good approximation to the perspective case would be to replace each Z_c with $\overline{Z}_c$ in equations (2.1.2):

$$x_c = f\frac{X_c}{\overline{Z}_c}, \quad y_c = f\frac{Y_c}{\overline{Z}_c} \quad . \tag{2.3.6}$$

In the weak-perspective model we allow for an offset of the principal point and a scaling between the camera co-ordinates and pixels. Here we assume that the same scaling factor applies in both the x and y directions and also that the shear term in (2.1.6) α is equal to zero. The transformation from the point in space, in camera co-ordinates to the point in the image, in pixels is the intrinsic parameter matrix of a scaled orthographic or weak-perspective camera,

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} sk & 0 & 0 & o_x \\ 0 & sk & 0 & o_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} , \qquad (2.3.7)$$

where $k = f/Z_c$. Hartley and Zisserman [HZ00] make a distinction between the scaled orthographic and weak perspective by allowing the weak perspective camera to have a different scaling in the x and y directions. Other authors [Sha95, SZB95] do not make this distinction, as it is usual to make the assumption that the image has square pixels.

We can multiply this intrinsic parameter matrix by the extrinsic parameter matrix to obtain the scaled orthographic or weak perspective camera matrix:

$$M_{wp} = \begin{pmatrix} kr_{11} & kr_{12} & kr_{13} & -k(RT)_1 + o_x \\ kr_{21} & kr_{22} & kr_{23} & -k(RT)_2 + o_y \\ 0 & 0 & 0 & 1 \end{pmatrix} . \qquad (2.3.8)$$

The weak perspective projection can be thought of as an orthographic projection onto the plane $Z_c = \overline{Z}_c$ followed by a perspective projection onto the image plane.

2.3.3 Properties of the Affine Camera

If we look at the affine projection matrices M_{or} and M_{wp} we can see that they both have the first three entries in the last row equal to zero. This is characteristic of the affine matrix. The affine camera, M_{aff}, was introduced to generalise the orthographic, weak perspective and para-perspective cameras [MZ92, SZB95]. An affine camera has a projection matrix of the form:

$$M_{aff} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ 0 & 0 & 0 & 1 \end{pmatrix} . \qquad (2.3.9)$$

It can be seen from the form of equation (2.3.9) that an affine camera allows us to represent the pixels, x and y, as a linear combination of the world co-ordinates, X, Y and Z:

$$\begin{aligned} x &= q_{11}X + q_{12}Y + q_{13}Z + q_{14} \\ y &= q_{21}X + q_{22}Y + q_{23}Z + q_{24} \end{aligned} . \qquad (2.3.10)$$

These linear equations provide an approximation to the perspective camera that will be used in chapter 4 to determine linear relationships between views.

2.4 Estimating the Accuracy of the Affine Approximation

In developing the view synthesis method in chapters 4 and 5 we use the affine camera model as an approximation to the perspective model. When evaluating the method it is useful to know how appropriate it is to make this assumption. We introduce an affine invariant property that will be used in section 4.3.2 to assess from the images how close the imaging system is to being affine. Details of projective and affine invariants can be found in [Zis92]. Here we give a brief discussion of the affine invariants that will be used to evaluate how good the affine camera model is. There are affine invariants of four points on a plane, provided of course that no three of the points are collinear.

To understand how the affine invariants arise, we note that, given four points on a plane we are able to form two pairs of triangles. It is known that the ratios of the areas of these triangles remain invariant under 2D affine transformations in the plane. In what follows we will show that the invariants are also valid for 3D to 2D affine projections, but not for perspective projections. In order to do this we choose our world co-ordinate system such that the four points lie in the XY plane, that is they all have $Z \equiv 0$. It is always possible to redefine 3D points that lie on a single plane as 2D points by a rotation and translation of the axis. We know from equation (2.3.9) that the image points can be written as a linear combination of the world points. Since we have chosen our co-ordinate system such that the points on the plane have $Z \equiv 0$ we are able to write our image points (x_i, y_i) as:

$$\begin{aligned} x_i &= q_{11}X_i + q_{12}Y_i + q_{14} \\ y_i &= q_{21}X_i + q_{22}Y_i + q_{24} \end{aligned} \qquad (2.4.1)$$

Equation (2.4.1) can be written in matrix from as:

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = M_{aff} \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \quad , \qquad (2.4.2)$$

where M_{aff} is the affine projection matrix $\begin{pmatrix} q_{11} & q_{12} & q_{14} \\ q_{21} & q_{22} & q_{24} \\ 0 & 0 & 1 \end{pmatrix}$.

Given four points lying on a plane there are in fact two independent pairs of triangles that can be formed from the four points and hence we are able to form two invariants from four points. If we choose our four points lying on a plane to be p_1, p_2, p_3 and p_4, where $p_i^T = (x_i, y_i, 1)^T$, then the two invariants, I_1 and I_2 are defined by:

$$I_1 = \frac{|m_{123}|}{|m_{134}|} \quad , \qquad I_2 = \frac{|m_{124}|}{|m_{234}|} \quad , \tag{2.4.3}$$

where m_{ijk} is the 3×3 matrix (p_i, p_j, p_k) and $|m|$ is the determinant of m that will give twice the area of triangle $p_i\ p_j\ p_k$.

We will now show that the values of I_1 and I_2 remain invariant under affine projections. If we write out invariant I_1 in terms of the image co-ordinates we obtain:

$$\frac{\begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & x_3 & x_4 \\ y_1 & y_3 & y_4 \\ 1 & 1 & 1 \end{vmatrix}} = \frac{\left| M_{aff} \begin{pmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{pmatrix} \right|}{\left| M_{aff} \begin{pmatrix} X_1 & X_3 & X_4 \\ Y_1 & Y_3 & Y_4 \\ 1 & 1 & 1 \end{pmatrix} \right|} \quad . \tag{2.4.4}$$

We know from the properties of determinants [Lip91] that the determinant of a product of two matrices A and B is equal to the product of their determinants:

$$|AB| = |A||B| \quad . \tag{2.4.5}$$

If we apply this property to the right hand of side of (2.4.4) we obtain:

$$\frac{\begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & x_3 & x_4 \\ y_1 & y_3 & y_4 \\ 1 & 1 & 1 \end{vmatrix}} = \frac{|M_{aff}| \begin{vmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{vmatrix}}{|M_{aff}| \begin{vmatrix} X_1 & X_3 & X_4 \\ Y_1 & Y_3 & Y_4 \\ 1 & 1 & 1 \end{vmatrix}} \quad . \tag{2.4.6}$$

The determinant of the affine transformation matrix $|M_{aff}|$ provided it is non-zero cancels out in (2.4.6) and the result is the invariant property I_1:

$$\frac{\begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & x_3 & x_4 \\ y_1 & y_3 & y_4 \\ 1 & 1 & 1 \end{vmatrix}} = \frac{\begin{vmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} X_1 & X_3 & X_4 \\ Y_1 & Y_3 & Y_4 \\ 1 & 1 & 1 \end{vmatrix}} \quad . \tag{2.4.7}$$

To show that I_1 and I_2 are not invariant under perspective projections we note that we have included a scaling factor w in equation (2.1.12). Again we are able to choose our world co-ordinates such that the world points lie on the plane $Z \equiv 0$. Each image point (x_i, y_i) can be represented using matrix notation as:

$$\begin{pmatrix} w_i x_i \\ w_i y_i \\ w_i \end{pmatrix} = M \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \quad , \tag{2.4.8}$$

where M is the projection matrix $\begin{pmatrix} q_{11} & q_{12} & q_{14} \\ q_{21} & q_{22} & q_{24} \\ q_{31} & q_{32} & q_{34} \end{pmatrix}$ and $w_i = q_{31}X_i + q_{32}Y_i + q_{34}$ in general is not equal to one and is not constant as it depends on X and Y.

If we look at the ratios of the determinants given in I_1 we obtain:

$$\frac{\begin{vmatrix} w_1x_1 & w_2x_2 & w_3x_3 \\ w_1y_1 & w_2y_2 & w_3y_3 \\ w_1 & w_2 & w_3 \end{vmatrix}}{\begin{vmatrix} w_1x_1 & w_3x_3 & w_4x_4 \\ w_1y_1 & w_3y_3 & w_4y_4 \\ w_1 & w_3 & w_4 \end{vmatrix}} = \frac{\left| M \begin{pmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{pmatrix} \right|}{\left| M \begin{pmatrix} X_1 & X_3 & X_4 \\ Y_1 & Y_3 & Y_4 \\ 1 & 1 & 1 \end{pmatrix} \right|} \quad . \tag{2.4.9}$$

We can again cancel out the projection matrix M on the right side in (2.4.9). We can also rearrange the left side by separating out the scaling factors w_i into a diagonal matrix:

$$\frac{\left| \begin{pmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \right|}{\left| \begin{pmatrix} w_1 & 0 & 0 \\ 0 & w_3 & 0 \\ 0 & 0 & w_4 \end{pmatrix} \begin{pmatrix} x_1 & x_3 & x_4 \\ y_1 & y_3 & y_4 \\ 1 & 1 & 1 \end{pmatrix} \right|} = \frac{\begin{vmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} X_1 & X_3 & X_4 \\ Y_1 & Y_3 & Y_4 \\ 1 & 1 & 1 \end{vmatrix}} \quad . \tag{2.4.10}$$

Since the determinant of a diagonal matrix is the product of its entries, we can re-write (2.4.10) as:

$$\frac{w_1 w_2 w_3 \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix}}{w_1 w_3 w_4 \begin{vmatrix} x_1 & x_3 & x_4 \\ y_1 & y_3 & y_4 \\ 1 & 1 & 1 \end{vmatrix}} = \frac{\begin{vmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} X_1 & X_3 & X_4 \\ Y_1 & Y_3 & Y_4 \\ 1 & 1 & 1 \end{vmatrix}} \quad . \tag{2.4.11}$$

Hence we can see that the invariant I_1 does not hold in general in the perspective case since $w_2 \neq w_4$. In order to find a perspective invariant we need to multiply by the ratio of another pair of determinants such that the scaling factors on the left hand side all cancel. It thus turns out [MZ92], that a perspective invariant can be constructed from five points on a plane (with no three collinear) by forming the ratios of ratios of the areas of triangles or determinants. Again, there are two distinct invariants which may be defined as [MZ92]:

$$I_{p1} = \frac{|m_{123}||m_{145}|}{|m_{134}||m_{125}|}, \quad \text{and} \quad I_{p2} = \frac{|m_{134}||m_{245}|}{|m_{234}||m_{145}|} \quad . \tag{2.4.12}$$

2.5 The Geometry of Two Affine Views

In this section we introduce the epipolar geometry and fundamental matrix for the case of affine cameras [HZ00, SZB95]. Using the affine camera as an approximation to the perspective camera has the advantage of leading to more stable algorithms and of simplifying the mathematics used. Here we show that the affine fundamental matrix can be represented as a linear combination of the co-ordinates in the two views.

2.5.1 Affine Epipolar Geometry

The fundamental difference between the affine and perspective epipolar geometry is that in the affine case all epipolar lines are parallel. If we consider the affine camera as a perspective camera with its optical centre on the plane at infinity, then there is an infinite distance between the camera centre and the image plane, which means that all

the lines of projection will be parallel as illustrated in figure 2.4. We note that when moving the camera centre to infinity we do not need to keep the lines of projection perpendicular to the image plane. In the case where they are perpendicular and the intrinsic parameters define a similarity transformation the resulting projection is orthographic or weak perspective. However in the case of a general affine projection matrix the lines of projection will still be parallel although they will not necessarily be perpendicular to the image plane. If we consider two points in one image, p_1 and p_2 then, as usual, the corresponding points in the second image lie on the epipolar lines l_1 and l_2. As usual, these epipolar lines are images of the lines in space passing through the first camera centre and the respective image points p_1 and p_2; that is, they are lines of projection. Since under affine imaging the projection rays are parallel, the epipolar lines are also parallel.

Furthermore since the epipolar lines are parallel to each other and all epipolar lines pass through the epipole, then the epipole must be at infinity.

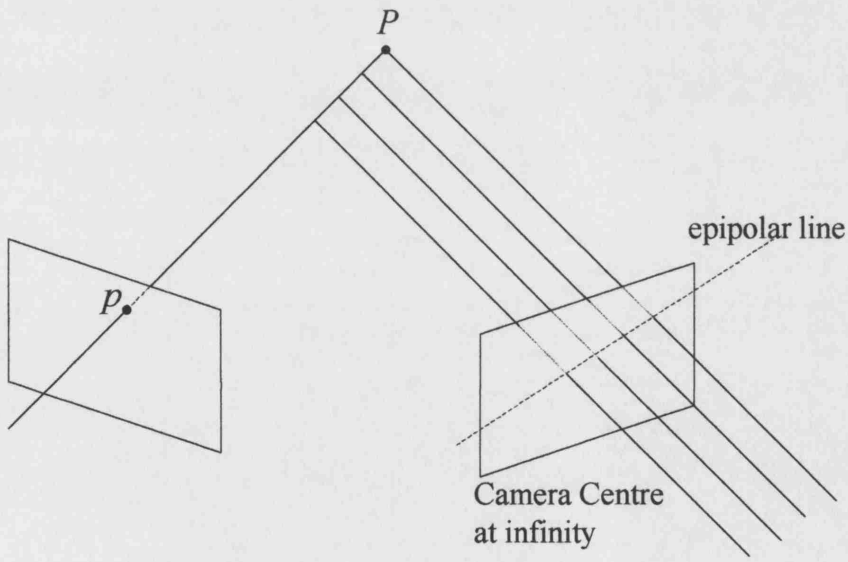


Figure 2.4. Affine Epipolar Geometry.

2.5.2 Affine Fundamental Matrices

The affine fundamental matrix [HZ00, Zis92] provides a relationship between corresponding points in two affine images in the same way that the fundamental matrix provides a relationship between corresponding points in two perspective images. Given the affine fundamental matrix and a point in one image it is possible to locate the epipolar line in the second image on which the corresponding point lies. We will now show that the affine fundamental matrix is equivalent to a linear relationship between co-ordinates of the corresponding points. Recall from equations (2.3.9) that in general, under any affine projection, the image co-ordinates of a point p can be written as a linear combination of the world co-ordinates of the point P. If we suppose that P is being imaged by two affine cameras to points p and p' then we can write the co-ordinates of these points as:

$$\begin{aligned} x &= q_{11}X + q_{12}Y + q_{13}Z + q_{14} \\ y &= q_{21}X + q_{22}Y + q_{23}Z + q_{24} \end{aligned} \quad , \tag{2.5.1}$$

and

$$\begin{aligned} x' &= q'_{11}X + q'_{12}Y + q'_{13}Z + q'_{14} \\ y' &= q'_{21}X + q'_{22}Y + q'_{23}Z + q'_{24} \end{aligned} \quad . \tag{2.5.2}$$

This provides us with four equations that are linear in X, Y and Z. In general, we can use three of these equations to obtain linear expressions for X, Y and Z in terms of x, y and x' say. We can then substitute for X, Y and Z in the last equation to obtain a linear relationship between x, y, x' and y':

$$ax + by + cx' + dy' + e = 0 \quad . \tag{2.5.3}$$

By using the homogeneous co-ordinate notation for the pixel co-ordinates, we can represent (2.5.3) as a matrix:

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & c \\ 0 & 0 & d \\ a & b & e \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \quad . \tag{2.5.4}$$

This gives the general form of the affine fundamental matrix, $F_{aff} = \begin{pmatrix} 0 & 0 & c \\ 0 & 0 & d \\ a & b & e \end{pmatrix}$.

2.6 Homographies

In chapter 1 we made the assumption that the scene was made up of planar surfaces and that these planar surfaces can be used to obtain dense correspondence between images. Here we show that there are mappings between corresponding planar surfaces in two images. Given that we are able to determine plane to plane mappings then, for any point lying on the plane in the first image, we are able to locate the corresponding point in the second image.

We know from section 2.4 that if we have a set of points in space that lie on a plane it is possible to transform the world co-ordinate system such that these points lie on the plane $Z \equiv 0$. The projection that maps these points to the image can then be represented as a 3×3 matrix:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong Q \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} & q_{14} \\ q_{21} & q_{22} & q_{24} \\ q_{31} & q_{32} & q_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad . \tag{2.6.1}$$

It then follows that, if we have two images of a planar surface, there is a mapping between the two imaged surfaces. For example, suppose we have a second image then there is a second projection matrix that maps the points in space onto the image points (x', y'):

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \cong Q' \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} q'_{11} & q'_{12} & q'_{14} \\ q'_{21} & q'_{22} & q'_{24} \\ q'_{31} & q'_{32} & q'_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad . \tag{2.6.2}$$

If we eliminate $(X, Y, 1)^T$ from (2.6.1) and (2.6.2) we may obtain a mapping between two images of points that lie on the planar surface provided that the inverse of the matrices Q and Q' exist:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} q_{11} & q_{12} & q_{14} \\ q_{21} & q_{22} & q_{24} \\ q_{31} & q_{32} & q_{34} \end{pmatrix} \begin{pmatrix} q'_{11} & q'_{12} & q'_{14} \\ q'_{21} & q'_{22} & q'_{24} \\ q'_{31} & q'_{32} & q'_{34} \end{pmatrix}^{-1} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$
$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = H \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \tag{2.6.3}$$

This plane to plane mapping, H, is known as a homography. The homography matrix H has eight degrees of freedom; there are nine entries but since we are working in

homogeneous co-ordinates the overall scale is unimportant. This means that we are able to determine the matrix H from correspondence of four points lying in a plane.

2.6.1 Affine Homographies

In an analogous way to the perspective case, we can also define homographies in the affine case. Recall from equation 2.4.1 that, in an appropriate frame of reference, points that lie on a plane in space that are being imaged with an affine camera can be written as:

$$\begin{aligned} x &= q_{11}X + q_{12}Y + q_{14} \\ y &= q_{21}X + q_{22}Y + q_{24} \end{aligned} \quad . \tag{2.6.4}$$

Given a second image we can express the points using a similar expression:

$$\begin{aligned} x' &= q'_{11}X + q'_{12}Y + q'_{14} \\ y' &= q'_{21}X + q'_{22}Y + q'_{24} \end{aligned} \quad . \tag{2.6.5}$$

If we eliminate X and Y from equations (2.6.4) and (2.6.5) we obtain a mapping between two images of points lying on the plane, taken under affine conditions:

$$\begin{aligned} x &= h_{11}x' + h_{12}y' + h_{13} \\ y &= h_{21}x' + h_{22}y' + h_{23} \end{aligned} \quad . \tag{2.6.6}$$

It is simply an affine (linear) transformation in the image plane. We can represent this affine planar mapping as a matrix using homogeneous co-ordinates, as:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \quad . \tag{2.6.7}$$

The affine homography only has six degrees of freedom, so it can be estimated from correspondence of three points lying in a plane.

Chapter 3

Multi-View Relationships and Encoding

In this chapter we discuss the relationships between multiple views of a scene. The existing literature on the multi-view relationships is written using a largely mathematical specialised language. In order to make the idea of multi-view relationships more accessible to the reader we begin in section 3.1 by describing some special simplified cases, where it is easy to derive the relationships between three views. Moving on to the general case, we give a geometric approach to defining the relationships between three views.

Once we have shown that there are relationships between views we move on to the literature review. We discuss the literature on the perspective relationships in section 3.2 and the affine relationships in section 3.3. In section 3.2 and 3.3 the relationships are derived from the camera matrices. In general we are working with uncalibrated images and the camera parameters will be unknown. Section 3.4 discusses methods for estimating these relationships directly from corresponding points across the images. In section 3.5 we discuss the encoding of a new view. Finally, we end by highlighting what is missing from existing work and what this thesis contributes.

3.1 Algebraic Relationships between Multiple Views

Much of the science of computer vision, in particular the geometrical relationships between a scene and images of it originates from photogrammetry [Atk96]. In photogrammetry, images of a scene are usually taken using calibrated cameras. These images are then used to obtain accurate information about the structure of the scene, which can be used to generate a 3D model of the scene. If desired, points in the scene could then be projected into a new view to obtain an image of the scene from a new

viewing position. This means that it is possible to find a relationship from points in images to the world co-ordinates and then a mapping from the world co-ordinates to a set of points in a new image. It follows therefore that it should be possible to find a relationship between corresponding points in the images without having first to recover the 3D information. Longuet-Higgins [L-H81] was the first to introduce the idea of multi-view relationships to the field of computer vision. He determined the essential matrix, which provides a relationship between two calibrated images of an object.

We have previously defined the two-view relationships in chapter 2. In this chapter we are concerned with multi-view relationships, i.e., between 3 or more images. While these relationships exist for any configuration of cameras it is helpful to see how these relationships arise in special cases before considering the general case. Three such configurations are described below. In the first case the views are obtained using a camera that is constrained to move perpendicular to its optical axis, in the direction of the X axis, say. This configuration leads to linear relationships between the views and enables us to make contact with some methods for view interpolation in the literature. In the second case the camera is translated along its optical axis, in the Z direction, which leads to bilinear relationships. Finally, the third special case we describe is where all three cameras share a common image plane and the camera centres lie on a plane parallel to this image plane. We will show how that this configuration leads to the general case giving trilinear relationships between the views. In this way we are able to show that unless the three camera centres lie on a line in space, it is possible to derive an algebraic operator, the trifocal tensor, as a combination of the three fundamental matrices that exist between pairs of the three views.

3.1.1 Three Cameras Positioned on a Baseline

We will now consider our first case where the camera is moved along a baseline parallel to the X axis. A translation of the camera is equivalent to a translation of the scene in the opposite direction. Consider an object in space with points on that object having co-ordinates $(X_0, Y_0, Z_0)^T$. The object is then translated in the direction of X_0 to three different positions such that the new co-ordinates are

$(X,Y,Z)^T$, $(X',Y',Z')^T$ and $(X'',Y'',Z'')^T$. Although it may seem redundant not to use X_0 as one of the viewpoints, by using X, X' and X'' as the viewpoints we are able to treat all three in a completely symmetrical fashion. The X co-ordinates are related to the original position of the object by:

$$\begin{aligned} X &= X_0 + T \\ X' &= X_0 + T' \\ X'' &= X_0 + T'' \end{aligned} \quad . \tag{3.1.1}$$

Since the object is only translated in the X direction the Y and Z co-ordinates remain constant:

$$\begin{aligned} Y &= Y' = Y'' = Y_0 \\ Z &= Z' = Z'' = Z_0 \end{aligned} \quad . \tag{3.1.2}$$

At each of the positions, the object is projected onto the image plane using the perspective projection equations given in chapter 2,

$$x = f\frac{X}{Z} \quad , \quad \text{and} \quad y = f\frac{Y}{Z} \quad . \tag{3.1.3}$$

The co-ordinates of the image points (x,y), (x',y') and (x'',y'') may then be written as:

$$\begin{aligned} x &= f\frac{X_0+T}{Z_0} & x' &= f\frac{X_0+T'}{Z_0} & x'' &= f\frac{X_0+T''}{Z_0} \\ y &= f\frac{Y_0}{Z_0} & y' &= f\frac{Y_0}{Z_0} & y'' &= f\frac{Y_0}{Z_0} \end{aligned} \quad . \tag{3.1.4}$$

We can see immediately that the y co-ordinate is the same in each image

$$y = y' = y'' \quad . \tag{3.1.5}$$

In this particularly simple case, equation (3.1.5) merely expresses the fact that the epipolar lines are all parallel to each other and to the baseline. To obtain a relationship between the x co-ordinates we need to eliminate X_0 and Z_0 from the equations in (3.1.4). If we subtract x' from x and x'' from x we obtain the following two equations:

$$x' - x = \frac{f}{Z_0}(T' - T), \qquad x'' - x = \frac{f}{Z_0}(T'' - T) \quad . \tag{3.1.6}$$

In order to eliminate the term f/Z_0 we can, provided $x'' \neq x$, divide $x' - x$ by $x'' - x$, to obtain:

$$\frac{x'-x}{x''-x}=\frac{T'-T}{T''-T} \quad . \tag{3.1.7}$$

By rearranging equation (3.1.7) we obtain a linear relationship between the x co-ordinates. Therefore, in the case where the object (or camera) is translated in the direction of the X axis the multi-view relationships are:

$$\begin{aligned} x &= \frac{T-T''}{T'-T''}x' - \frac{T-T'}{T'-T''}x'' \\ y &= y' = y'' \end{aligned} \quad . \tag{3.1.8}$$

We have derived the relationships between three views of an object when an ideal pinhole camera is only allowed to move in a single direction perpendicular to its optic axis. Since, by assumption T, T' and T'' are all different, (3.1.8) may be multiplied out and rearranged to give the more symmetrical form:

$$\begin{aligned} x(T'-T'')+x'(T''-T)+x''(T-T') &= 0 \\ y = y' &= y'' \end{aligned} \quad , \tag{3.1.9}$$

in which, in each term we cycle through the co-ordinates, dashed co-ordinates and double dashed co-ordinates. We can see that the relationships in (3.1.9) are linear. This is a result of the constraints we put on the viewing positions. We will now give another example of the relationships between three views where the movement of the object is again constrained, but to be along the optic axis, Z.

3.1.2 Translation in the Direction of the *Z* axis

As noted above, in this section we will restrict the movement of the object (or camera) to a translation in the direction of the Z axis. We continue to use the same notation as in the previous example for the world and image co-ordinates. We can thus write the Z co-ordinate for the three different positions of the object in terms of the initial position, as:

$$\begin{aligned} Z &= Z_0 + V \\ Z' &= Z_0 + V' \\ Z'' &= Z_0 + V'' \end{aligned} \quad . \tag{3.1.10}$$

The X and Y co-ordinates remain constant at each position of the object, so:

$$\begin{aligned} X &= X' = X'' = X_0 \\ Y &= Y' = Y'' = Y_0 \end{aligned} \quad . \tag{3.1.11}$$

Once again, we write down the expressions for the image co-ordinates:

$$x = f\frac{X_0}{Z_0 + V} \quad x' = f\frac{X_0}{Z_0 + V'} \quad x'' = f\frac{X_0}{Z_0 + V''}$$
$$y = f\frac{Y_0}{Z_0 + V} \quad y' = f\frac{Y_0}{Z_0 + V'} \quad y'' = f\frac{Y_0}{Z_0 + V''} \qquad (3.1.12)$$

If each of the first three equations in (3.1.12) is divided by the corresponding equation from the second three, we immediately see:

$$\frac{x}{y} = \frac{x'}{y'} = \frac{x''}{y''} \qquad (3.1.13)$$

These equations, independent of the object co-ordinates and of the translations V, V' and V'', are the equations of the epipolar lines, each of which passes through the centre of the image at $(0,0)$. They express the familiar fact that as a camera is moved along its optic axis the image points move radially. By eliminating the terms fX_0 from (3.1.12) we find that:

$$x(Z_0 + V) = x'(Z_0 + V')$$
$$x(Z_0 + V) = x''(Z_0 + V'') \qquad (3.1.14)$$

If we then re-arrange equations (3.1.14) we obtain:

$$Z_0(x - x') = x'V' - xV \qquad (3.1.15)$$

and

$$Z_0(x - x'') = x''V'' - xV \qquad (3.1.16)$$

Provided $x \neq x'$ and $x \neq x''$, we can then eliminate the term Z_0 by dividing (3.1.15) by (3.1.16), to give

$$\frac{x - x'}{x - x''} = \frac{x'V' - xV}{x''V'' - xV} \qquad (3.1.17)$$

If we now re-arrange equation (3.1.17) we obtain a bilinear relationship between the x co-ordinates of the three images which may be written in the following form:

$$xx'(V - V') + x'x''(V' - V'') + x''x(V'' - V) = 0 \qquad (3.1.18)$$

in which, once again we cycle through the co-ordinates, dashed co-ordinates and double dashed co-ordinates. If, instead of eliminating fX_0 and Z_0, we eliminate fY_0 and Z_0 from (3.1.12), we obtain a similar relationship to (3.1.18) in terms of the y co-ordinates of the three images. In this case, we obtain a pair of multi-view relationships:

$$xx'(V-V')+x'x''(V'-V'')+x''x(V''-V)=0$$
$$yy'(V-V')+y'y''(V'-V'')+y''y(V''-V)=0 \qquad (3.1.19)$$

The two equations (3.1.18) and (3.1.19) are the pair of relationships between the image co-ordinates of three views when the movement of the camera is constrained to a translation parallel to the optic axis Z of the camera. The fact that the equations are bilinear arises from the special configuration that we have used. This will not be the case for a general configuration of cameras. If we look at the pairs of equations in (3.1.9) and (3.1.19) we see that they refer to the x and y co-ordinates in separate equations. This will also not be the case when the viewpoints are in general positions. When equations (3.1.19) are taken together with the two equations for the epipolar lines (3.1.13), we appear to have four constraints on the image co-ordinates in the three views. This is more than we should expect to obtain from the six perspective equations on eliminating the three object co-ordinates X_0, Y_0 and Z_0. The situation is easily reconciled however, by noting that, given the epipolar line equations (3.1.13) either of the equations in (3.1.19) may (in general) be derived from the other. One way to see this is to divide the first equation in (3.1.19) by xx' and then to use the epipolar line equations to substitute for x''/x' and x''/x.

3.1.3 Three Cameras in a Plane

We will now consider a third special case. We extend the example given in section 3.1.1 allowing translations of the camera in the directions of both the X and Y axes. All cameras now have identical focal lengths and are arranged such that they share a common image plane. This image plane is parallel to the plane passing through the three camera centres. The three cameras are imaging a point P in space.

The 3D point P has co-ordinates (X_0,Y_0,Z_0) and we will call our image co-ordinates (x_0,y_0), (x_0',y_0') and (x_0'',y_0''). The camera is moved from its initial position to three different locations in the (X,Y) plane. The three camera positions are located by shifting the camera from its initial position by distances of T, T' and T'' in the direction of the X axis and distances of U, U' and U'' in the direction of the Y axis. We are assuming that the cameras share a common image plane, and we wish to express each of the co-ordinates of the three views with reference to the same co-

ordinate system within this common image plane. Therefore, if the camera is shifted in the (X,Y) by (T,U) then the image points (x_0,y_0) will also be shifted in the common image plane by (T,U). The projection equations for the three sets of image co-ordinates can therefore be written as:

$$x_0+T=\frac{f(X_0+T)}{Z_0} \quad \text{(a)} \qquad y_0+U=\frac{f(Y_0+U)}{Z_0} \quad \text{(b)}$$
$$x_0'+T'=\frac{f(X_0+T')}{Z_0} \quad \text{(c)}, \qquad y_0'+U'=\frac{f(Y_0+U')}{Z_0} \quad \text{(d)} \qquad . \qquad (3.1.20)$$
$$x_0''+T''=\frac{f(X_0+T'')}{Z_0} \quad \text{(e)} \qquad y_0''+U''=\frac{f(Y_0+U'')}{Z_0} \quad \text{(f)}$$

We have six equations in (3.1.20) and in order to determine the multi-view relationships we need to eliminate the world co-ordinates X_0, Y_0 and Z_0. If we subtract (a) from (c) in (3.1.20) we can eliminate X_0 to obtain:

$$x_0'-x_0+T'-T=\frac{f}{Z_0}(T'-T) \qquad . \qquad (3.1.21)$$

We could also use the pair (e) and (a) or the pair (e) and (c) in (3.1.20) to eliminate X_0 and obtain equations:

$$\begin{aligned} x_0''-x_0+T''-T&=\frac{f}{Z_0}(T''-T) \\ x_0''-x_0'+T''-T'&=\frac{f}{Z_0}(T''-T') \end{aligned} \qquad . \qquad (3.1.22)$$

Similarly we can use pairs of (b), (d) and (e) in (3.1.20) to eliminate Y_0 and obtain the equations:

$$\begin{aligned} y_0'-y_0+U'-U&=\frac{f}{Z_0}(U'-U) \\ y_0''-y_0+U''-U&=\frac{f}{Z_0}(U''-U) \\ y_0''-y_0'+U''-U'&=\frac{f}{Z_0}(U''-U') \end{aligned} \qquad . \qquad (3.1.23)$$

By eliminating the term f/Z_0 from equations (3.1.21), (3.1.22) and (3.1.23) we can see that:

$$\frac{x_0'-x_0}{T'-T}=\frac{x_0''-x_0}{T''-T}=\frac{x_0''-x_0'}{T''-T'}=\frac{y_0'-y_0}{U'-U}=\frac{y_0''-y_0}{U''-U}=\frac{y_0''-y_0'}{U''-U'} \qquad . \qquad (3.1.24)$$

There are five equations in (3.1.24). Only three of the five are independent because we started with six independent equations in (3.1.20) and have eliminated the terms in three of the variables, X_0, Y_0 and Z_0. This means that there are three relationships between the three views, which is what we would expect in the general case. We know that between two views of an object there exists a fundamental matrix that provides a relationship between the views. If we have three views then we can form three pairs of such relationships.

One choice of an independent subset of equations of (3.1.24) is equivalent to the three fundamental matrices. These fundamental matrices are the three pairs of relationships in (3.1.24) that only involve the image co-ordinates of two of the views. These relationships are:

$$\frac{x_0' - x_0}{T' - T} = \frac{y_0' - y_0}{U' - U}, \quad \frac{x_0'' - x_0}{T'' - T} = \frac{y_0'' - y_0}{U'' - U} \text{ and } \frac{x_0'' - x_0'}{T'' - T'} = \frac{y_0'' - y_0'}{U'' - U'} \quad . \tag{3.1.25}$$

However we can also choose three independent relationships from (3.1.24) that involve the co-ordinates of all three views, thereby generating multi-view relationships. Thus we may choose our independent subset to be:

$$\frac{x_0' - x_0}{T' - T} = \frac{x_0'' - x_0}{T'' - T} = \frac{y_0' - y_0}{U' - U} = \frac{y_0'' - y_0}{U'' - U} \quad . \tag{3.1.26}$$

If we re-arrange the first equation of (3.1.26) we obtain:

$$a_0 x_0 + a_1 x_0' + a_2 x_0'' = 0 \quad , \tag{3.1.27}$$

where the a_i are constant terms, $a_0 = T' - T''$, $a_1 = T'' - T$ and $a_2 = T - T'$. Equation (3.1.27) is the first of our three-view relationships and is the same relationship obtained in equation (3.1.9). We will now show that, in this case, it is possible to express this relationship as a linear combination of the three fundamental matrices. In order to do this we re-write our fundamental matrices from equation (3.1.25) as:

$$\begin{aligned} (U' - U)(x_0' - x_0) &= (T' - T)(y_0' - y_0) \quad \text{(a)} \\ (U'' - U)(x_0'' - x_0) &= (T'' - T)(y_0'' - y_0) \quad \text{(b)} \\ (U'' - U')(x_0'' - x_0') &= (T'' - T')(y_0'' - y_0') \quad \text{(c)} \end{aligned} \quad . \tag{3.1.28}$$

Since we are looking for a relationship between the x co-ordinates we need to eliminate the terms in the y co-ordinates. If we divide equations (a), (b) and (c) by the terms $T' - T$, $T'' - T$ and $T'' - T'$ respectively we obtain expressions involving only the y co-ordinates on the right hand sides of the equations in (3.1.28).

$$\begin{aligned}
&\frac{U'-U}{T'-T}(x_0'-x_0)=y_0'-y_0 \quad \text{(a)}\\
&\frac{U''-U}{T''-T}(x_0''-x_0)=y_0''-y_0 \quad \text{(b)}\\
&\frac{U''-U'}{T''-T'}(x_0''-x_0')=y_0''-y_0' \quad \text{(c)}
\end{aligned} \qquad . \qquad (3.1.29)$$

We can now eliminate the y co-ordinates from equation (3.1.29) by subtracting (b) from (a) and adding (c). This leaves a zero term on the right hand side and we are left with a relationship between the x co-ordinates:

$$\frac{U'-U}{T'-T}(x_0'-x_0)-\frac{U''-U}{T''-T}(x_0''-x_0)+\frac{U''-U'}{T''-T'}(x_0''-x_0')=0 \quad . \qquad (3.1.30)$$

By re-arranging (3.1.30) we obtain:

$$\begin{aligned}
&\left(\frac{U''-U}{T''-T}-\frac{U'-U}{T'-T}\right)x_0+\left(\frac{U'-U}{T'-T}-\frac{U''-U'}{T''-T'}\right)x_0'\\
&\qquad+\left(\frac{U''-U'}{T''-T'}-\frac{U''-U}{T''-T}\right)x_0''=0
\end{aligned} \qquad . \qquad (3.1.31)$$

An important point to notice at this point is that if the ratio $(U'-U)/(T'-T)$ is equal to the ratio of $(U''-U)/(T''-T)$ then the three co-efficients in (3.1.31) are all equal to zero. The case where the two ratios are equal corresponds to the case where the three cameras lie on a line. It is well known that the trifocal tensor cannot be determined from the three fundamental matrices in this case [HZ00]. This problem is also seen in the example in section 3.1.1, where the trifocal relationships in equation (3.1.9) cannot be expressed by using the fundamental matrices in equation (3.1.5). If we now assume that the cameras do not lie on a line and re-arrange and simplify the co-efficients in equation (3.1.31) we obtain:

$$(T'-T'')x_0+(T''-T)x_0'+(T-T')x_0''=0 \quad . \qquad (3.1.32)$$

It can be seen that equation (3.1.32), which we derived as a combination of the three fundamental matrices, is the same as our three-view relationship (3.1.27).

In order to obtain a second multi-view relationship we can re-arrange the equation in (3.1.26) that involves the y co-ordinates of the three views:

$$b_0y_0+b_1y_0'+b_2y_0''=0 \quad , \qquad (3.1.33)$$

where $b_0 = U' - U''$, $b_1 = U'' - U$ and $b_2 = U - U'$. This equation can also be derived as a combination of the three fundamental matrices by eliminating the x co-ordinates of the three views, provided of course that $\frac{(U'-U)}{(T'-T)} \neq \frac{(U''-U)}{(T''-T)}$.

When the multi-view relationships are used to synthesise a view, two relationships are needed in order to recover the x and y co-ordinates of the point in the target view. We stated earlier that there are in fact three independent relationships in equation (3.1.26). One possible choice for the third independent equation is:

$$\frac{x_0'' - x_0}{T'' - T} = \frac{y_0'' - y_0}{U'' - U} \quad . \tag{3.1.34}$$

Any other equation will be dependent on the three multi-view relationships that we have chosen, (3.1.27), (3.1.33) and (3.1.34). Although only two relationships are needed to synthesise a view, the third equation can be used when one of the co-efficients in (3.1.27) or (3.1.33) is equal to zero. If we consider the first view with co-ordinates (x_0, y_0) to be our target view to be synthesised, then in the case where $T'' = T'$, a_0 is equal to zero in equation (3.1.27) and the x_0 co-ordinate cannot be recovered using this equation. In this case it is possible to use equation (3.1.34) instead provided, of course $T'' \neq T$.

If we assume for now that $T'' \neq T'$ then, in this special case, where the cameras share a common image plane parallel to the plane defined by the camera centres, one possible pair of multi-view relationships is provided by equations (3.1.27) and (3.1.33). We have seen that this pair of equations is not unique. There are three independent relationships, which corresponds to the fact that there are three fundamental matrices. In this special case, these equations are linear.

We have given the special cases as an introduction to the relationships between three views and to motivate us to derive the general case. We do not include the special case of a general 3D translation of the three cameras because the 2D translation can be easily extended to the general case. We will now show that the relationships between three arbitrary views are, in general, trilinear.

3.1.4 Three Cameras in General Position

Given three cameras in general position we are able to define a unique plane in space that passes through the three camera centres. We can then choose another plane in space that is parallel to the plane defined by the three camera centres. This new plane can be thought of as a common image plane for the three views and we can project all the points in space onto this common image plane using each of the cameras. This is equivalent to what we have done in the previous example. Each point in space, P, projects to three points in the common image plane, (x_0, y_0), (x_0', y_0') and (x_0'', y_0''). There are planar mapping (homographies) between this common image plane and the actual image planes of the three cameras. By using homogeneous co-ordinates we can represent these homographies by 3×3 matrices. Thus, we will represent the mapping between the common image plane (x_0, y_0) and the actual image plane (x, y) of the first camera by a matrix H:

$$\begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad . \tag{3.1.35}$$

Although this is a mapping of the whole (x, y) plane we only use it to map points from the common image plane that were projected onto this plane by the first camera.

We may also to define similar mappings, H' and H'', between the actual image planes of the second and third cameras respectively and the common image plane:

$$\begin{pmatrix} x_0' \\ y_0' \\ 1 \end{pmatrix} = H' \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}, \qquad \begin{pmatrix} x_0'' \\ y_0'' \\ 1 \end{pmatrix} = H'' \begin{pmatrix} x'' \\ y'' \\ 1 \end{pmatrix} \quad . \tag{3.1.36}$$

By carrying out the matrix-vector multiplications in equations (3.1.35) and (3.1.36) we obtain equations for the co-ordinates (x_0, y_0), (x_0', y_0') and (x_0'', y_0'') in terms of the actual camera co-ordinates (x, y), (x', y') and (x'', y'') respectively.

$$x_0 = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \qquad y_0 = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$
$$x'_0 = \frac{h'_{11}x' + h'_{12}y' + h'_{13}}{h'_{31}x' + h'_{32}y' + h'_{33}}, \qquad y'_0 = \frac{h'_{21}x' + h'_{22}y' + h'_{23}}{h'_{31}x' + h'_{32}y' + h'_{33}} \quad . \qquad (3.1.37)$$
$$x''_0 = \frac{h''_{11}x'' + h''_{12}y'' + h''_{13}}{h''_{31}x'' + h''_{32}y'' + h''_{33}} \qquad y''_0 = \frac{h''_{21}x'' + h''_{22}y'' + h''_{23}}{h''_{31}x'' + h''_{32}y'' + h''_{33}}$$

If we then substitute (3.1.37) into equations (3.1.27) and (3.1.33) we obtain:

$$a_0 \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} + a_1 \frac{h'_{11}x' + h'_{12}y' + h'_{13}}{h'_{31}x' + h'_{32}y' + h'_{33}} + a_2 \frac{h''_{11}x'' + h''_{12}y'' + h''_{13}}{h''_{31}x'' + h''_{32}y'' + h''_{33}} = 0 \quad , \quad (3.1.38)$$

and

$$b_0 \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} + b_1 \frac{h'_{21}x' + h'_{22}y' + h'_{23}}{h'_{31}x' + h'_{32}y' + h'_{33}} + b_2 \frac{h''_{21}x'' + h''_{22}y'' + h''_{23}}{h''_{31}x'' + h''_{32}y'' + h''_{33}} = 0 \quad , \quad (3.1.39)$$

It can be seen that if we multiply out the denominators in equations (3.1.38) and (3.1.39) above then we obtain a pair of trilinear equations. These equations are a pair of relationships between three arbitrary views of an object. Similarly we can substitute the expressions in (3.1.37) into our third multi-view relationship (3.1.34) to obtain a third independent trilinear relationship.

3.2 Perspective Relationships

One of the first relationships between a pair of views to be introduced into the field of computer vision was that provided by the essential matrix. As stated earlier this matrix was introduced by Longuet-Higgins [L-H81] and provides a relationship between two views taken from calibrated cameras. The fact that it was possible to generalise the essential matrix to the uncalibrated case was realised both by Faugeras [Fau92, FLM92] and Hartley [Har92]. The matrix obtained in the uncalibrated case is known as the fundamental matrix. This matrix was derived in chapter 2, section 2.2.3. A more detailed analysis of the fundamental matrix can be found in [LF96]. The relationships between three views can be expressed using tensor notation [Har97a, TZ97]. This is explored in section 3.3.1. Section 3.3.2 goes on to discuss the relationships between four views and the fact that, in principle, there is no new information about scene structure that can be determined by using more than three views [Tri95].

3.2.1 Three-View Relationships

As happened in the study of the relationships of two views, the relationships between three views were first determined for the case of calibrated cameras. Weng et al [WAH92] and Spetsakis and Aloimonos [SA90, Spe91] were among the first to produce such relationships between three calibrated views.

Hartley later showed that these relationships were also valid in the uncalibrated case [Har94b]. In this paper Hartley describes a method for determining scene structure from a set of matching lines in three images. Independently of Hartley's work, Shashua also introduced a set of trilinearity conditions for matching points in three images [Sha94]. Shashua's trilinearity conditions are reproduced in equation (3.2.1).

$$\begin{aligned} &x''(\alpha_1 x+\alpha_2 y+\alpha_3)+x''x'(\alpha_4 x+\alpha_5 y+\alpha_6)+ \\ &\qquad x'(\alpha_7 x+\alpha_8 y+\alpha_9)+\alpha_{10}x+\alpha_{11}y+\alpha_{12}=0 \\ &y''(\beta_1 x+\beta_2 y+\beta_3)+y''x'(\beta_4 x+\beta_5 y+\beta_6)+ \\ &\qquad x'(\beta_7 x+\beta_8 y+\beta_9)+\beta_{10}x+\beta_{11}y+\beta_{12}=0 \end{aligned} \qquad (3.2.1)$$

where $\alpha_i = \beta_i$ for $i = 1,...,6$. There are eighteen independent co-efficients in equations (3.2.1), this corresponds to the fact that the trifocal tensor has eighteen degrees of freedom [TZ96].

Hartley later provided an algorithm for determining the scene structure from either a set of matching point or lines or a mixture of both [Har97a, Har94a]. He also showed that the two sets of trilinear relationships given in [Har94b] and [Sha94] are equivalent [Har95a, Har97a] and can be expressed using tensor notation. This is known as the trifocal tensor. In [Har97a] the trifocal tensor is defined using elements of the projection matrices. In this approach, the first projection matrix, M, is defined such that it is of the form:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \qquad (3.2.2)$$

It is always possible to assume that M is of the form given in (3.2.2) since we are able to choose an arbitrary world co-ordinate system. The second and third camera projection matrices, M', and M'' are defined by:

$$M' = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}, \text{ and } \quad M'' = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{pmatrix} . \qquad (3.2.3)$$

It is then possible to define the trifocal tensor, T, using the elements of the projection matrices as [Har97a, TZ96]:

$$T_{ijk} = M'_{ji} M''_{k4} - M'_{j4} M''_{ki} \quad , \qquad (3.2.4)$$

where $i, j, k = 1,..,3$. It can be seen that the trifocal tensor is a rank three $3 \times 3 \times 3$ tensor and therefore has 27 elements. If we denote a matching point in the three views in homogeneous co-ordinates by $p = (p_1, p_2, p_3)^T$, $p' = (p'_1, p'_2, p'_3)^T$ and $p'' = (p''_1, p''_2, p''_3)^T$ respectively, then it is possible to use the trifocal tensor to express the following relationship between the points [TZ96]:

$$p''_l \cong p'_i \sum_{k=1}^{3} p_k T_{kjl} - p'_j \sum_{k=1}^{3} p_k T_{kil} \quad , \qquad (3.2.5)$$

where $i, j = 1,..,3$ and the symbol "$\cong$" represents equality up to a scaling factor.

Similarly, if we write corresponding lines in the three views as $l = (l_1, l_2, l_3)^T$, $l' = (l'_1, l'_2, l'_3)^T$ and $l'' = (l''_1, l''_2, l''_3)^T$ respectively, then the trifocal tensor provides the following relationship between the lines:

$$l_i = \sum_{j=1}^{3} \sum_{k=1}^{3} T_{ijk} l'_j l''_k \quad . \qquad (3.2.6)$$

The trifocal tensor has eighteen degrees of freedom. There are eleven degrees of freedom for each camera matrix from which we have to subtract fifteen degrees of freedom to allow for the fact that we can multiply each camera matrix by a 4×4 projective matrix and still obtain the same relationship between the views. The number of degrees of freedom remaining then is $33 - 15 = 18$. The trifocal tensor is defined up to a projective scaling which means that only the twenty-six ratios of the elements of T are important. Therefore are eight constraints on the twenty-seven elements of T (26-18=8).

3.2.2 Higher-Order Relationships

We have seen that the fundamental matrix provides a bilinear relationship between corresponding points in two images of a scene and the trifocal tensor provides a

trilinear relationship between three views. There is also a set of quadrilinear relationships between corresponding points in four views [Tri95]. This set of relationships is known as the quadrifocal tensor [Har98, SW00]. The quadrifocal tensor is not as widely used as the three-view and two-view relationships, one of the reasons being that it is greatly overparameterised [Har98, Hey98]. The quadrifocal relationships can be represented by a $3\times3\times3\times3$ tensor usually denoted by Q. This means that it has 81 elements, the scaling of which is unimportant. However, Q has 29 degrees of freedom, eleven for each of the four camera projection matrices minus fifteen since we can multiply each of the projection matrices by the same arbitraty projective transformation. This means that there are fifty-one constraints on the elements of Q, (80-29=51) [Hey98].

Given four projection matrices, M, M', M'' and M''', it is possible to write the quadrifocal tensor as [Har98, Hey98, Hey00]:

$$Q_{pqrs} = \det\begin{bmatrix} M_p \\ M'_q \\ M''_r \\ M'''_s \end{bmatrix} \qquad p,q,r,s = 1..3 \quad , \tag{3.2.7}$$

where M_a corresponds to the a^{th} row of the projection matrix M etc. If we then, as usual, let p, p', p'' and p''', in homogeneous co-ordinates, denote a corresponding point in each of the four images respectively, then the quadrifocal tensor provides a set of relationships between the points:

$$p_i p'_j p''_k p'''_l \varepsilon_{ipt}\varepsilon_{jqu}\varepsilon_{krv}\varepsilon_{lsw}Q_{pqrs} = 0_{tuvw} \quad , \tag{3.2.8}$$

where $t,u,v,w = 1..3$ and i,j,k,l,p,q,r,s are summed over the values 1 to 3. The permutation (Levi-Civita) "tensor" ε_{abc} is defined, as in [Hey98, Har98], such that $\varepsilon_{123} = \varepsilon_{231} = \varepsilon_{312} = 1$, $\varepsilon_{321} = \varepsilon_{132} = \varepsilon_{213} = -1$ and $\varepsilon_{abc} = 0$ if any two of the indices are the same. Similarly, we can use Q to write a relationship between corresponding lines in four views [Har95b, Hey98], l, l', l'' and l''':

$$l_p l'_q l''_r l'''_s Q_{pqrs} = 0 \quad , \tag{3.2.9}$$

where, as in the case of matching points, the indices p, q, r and s are summed over the range 1 to 3.

It can be seen from (3.2.8) that each point correspondence across the four images provides 81 equations in the elements of Q, of which only 16 are independent

[Hey98, Har95b]. Since we are only interested in the elements of Q up to a scale factor, it would appear that 5 corresponding points in the four images provides us with 80 equations which is enough to solve for its elements. However, it was shown by Hartley [Har95b], that the set of 16 independent equations obtained from one point correspondence across the four views will *not* be independent of the set of 16 equations provided by a second corresponding point across the four views. In fact 6 point correspondences are needed to solve for the elements of Q [Har95b, Hey98].

We have shown in section 3.1 that the relationships between three views (trifocal tensor) can be expressed as a linear combination of the two view relationships (fundamental matrices) between the constituent pairs of the views, except when the three views lie on a line in space. It turns out that the quadrifocal tensor can also be expressed as a linear combination of the lower order tensors as is shown in [FM95a]. Furthermore, this is always so, whatever the camera configuration. This means that, in principle, no new information about the structure of the scene can be obtained by using four views instead of three [FM95a, Tri95]. However, the last statement assumes that the relationships can be found exactly, i.e. that there are no errors on any of the points [SHB99]. In practice, when errors are present using a forth view could contribute by adding stability and allow a more accurate reconstruction [Har98].

3.2.3 A Common Framework for N Views

A common framework has been defined for expressing the multi-view relationships by Triggs [Tri95] and also by Faugeras and Mourrain [FM95a]. Briefly, we proceed as follows. Recall the standard camera equation, relating a point in space $P=(X,Y,Z,1)^T$ to a point in the image $p=(x,y,1)^T$:

$$\lambda p = MP \quad , \tag{3.2.10}$$

where M is a 3×4 projection matrix and λ is a constant scale factor. Suppose that we have m views of point P. We can then write the projection equations as:

$$\lambda_i p_i = M^i P \quad , \quad i=1..m \quad , \tag{3.2.11}$$

where M^i is the i^{th} projection matrix, p_i is image of point P in the i^{th} image and the λ_i are constant scale factors. It is then possible to represent the set of m projection

equations in (3.2.11) in the combined framework [FM95a, Tri95] by means of a $3m \times (m+4)$ matrix equation:

$$\underbrace{\begin{pmatrix} M^1 & p_1 & & 0 & . & . & 0 \\ M^2 & 0 & p_2 & 0 & . & . & 0 \\ M^3 & 0 & 0 & p_3 & . & . & 0 \\ . & . & . & . & . & & \\ . & . & . & . & & . & \\ M^m & 0 & 0 & 0 & . & . & p_m \end{pmatrix}}_{A} \begin{pmatrix} P \\ -\lambda_1 \\ -\lambda_2 \\ -\lambda_3 \\ . \\ . \\ -\lambda_m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} \quad . \qquad (3.2.12)$$

The relationships between the corresponding points in the m views can then be derived by finding the determinants of the $(m+4)\times(m+4)$ submatrices of the $(3m)\times(m+4)$ matrix A [Tri95]. The example where $m=4$, as shown by Heyden [Hey98, Hey00], leads to a derivation of the fundamental matrices, the trifocal tensors and the quadrifocal tensor and the relationships between the different types of tensors.

3.3 Affine Relationships

As we have noted several times the affine camera is widely used in computer vision as an approximation to the perspective camera model [AHKO99, FRM98, HTM99, MC98, QK96, Sha95, TK92 and UB91]. Koenderink and van Doorn used the assumption of orthographic cameras in the problem of determining the structure of objects from a sequence of images [KV91]. In this paper they assumed a small field of view. When the field of view is small and the variation in depth of the scene is small compared to the overall depth, the affine camera (which is a generalisation of the orthographic camera) is a good approximation to the perspective [SZB95]. Even when the viewing conditions include strong perspective effects, i.e. when the field of view and the variation in depth are not small, the affine approximation still allows us to synthesise realistic-looking novel views and has other advantages indicated below.

In particular, one advantage of using the affine camera is that it leads to a simplification of the mathematics, which results in the multi-view relationships being linear [UB91, Ull96]. Mathematically this means that they have fewer degrees of freedom and can be estimated from fewer corresponding points across the images than in the perspective case and that the algorithms are, in general, of a straightforward

linear least squares type. For example the affine fundamental matrix has four degrees of freedom and can be determined from four point correspondences whereas the perspective fundamental matrix has seven degrees of freedom and requires seven matching points [Zis92]. Similarly the affine trifocal tensor has at most twelve degrees of freedom [MC98, TM99], in contrast to the perspective trifocal tensor which has eighteen degrees of freedom [TZ97]. The number of degrees of freedom of the affine trifocal tensor can be reduced from twelve to nine by using centre-of-mass co-ordinates as will be shown in section 3.3.2.

From a practical point of view, more significantly, the affine relationships are more stable (less sensitive to errors) than the corresponding perspective relationships and the affine approximation has the advantage that it will continue to work, although it will give *inaccurate* results, even when perspective effects are large. This is unlike the perspective case, which leads to ambiguities and/or instabilities and unreliable results which can be far from correct when perspective effects are small and the viewing conditions are close to being affine [SZB95].

Zisserman and Mundy first introduced the affine camera to computer vision in 1992 [MZ92]. The affine camera as a projection model was introduced to generalise the orthographic, scaled orthographic and para-perspective models.

We have previously introduced the affine epipolar geometry and affine fundamental matrix in chapter 2. Construction of the epipolar lines is described in [DZB92]. The affine fundamental matrix was first defined as a matrix by Zisserman [Zis92]. A more detail description of the affine epipolar geometry and a derivation of the affine fundamental matrix can be found in Shapiro et al [SZB95], where the fundamental matrix is also represented as a linear relationship between matching co-ordinates in two views. This linear representation of the fundamental matrix is equation (2.5.3) given in chapter 2.

We have seen in section 2.3.3 that the assumption of an affine camera provides a linear relationship between the world co-ordinates and the image co-ordinates. It then follows that the affine multi-view relationships will also be linear in the image co-ordinates. It is possible to obtain the multi-view relationships by eliminating the world co-ordinates from a set of linear equations for each of the image points and thereby derive a linear relationship between the image co-ordinates. We have already seen that this is the case for the affine fundamental matrix in (2.5.3).

Ullman and Basri [UB91] used the assumption of an orthographic camera in object recognition. They showed that by doing so, it is possible to represent an image as a linear combination of other images. It is then possible to use a limited set of images to represent the appearance of an object and then to use linear combinations of these *basis views* to construct novel views of the object. New instances of the object in previously unseen images may therefore be detected by comparison with the novel views so constructed.

3.3.1 Relationships between Three Affine Views

In this section we introduce the relationships between three affine views of an object. First we derive the affine three-view relationships for a specific example where the camera is constrained to move in a circular path which lies in the plane $Y = 0$ described by the world co-ordinate system. We then go on to describe the general three-view relationships that exist between three affine views.

We begin by looking at the work of Ullman and Basri [UB91, Ull96]. First we consider the specific example, given in [UB91] and [Ull96], where the camera is rotated about the *Y*-axis of the world co-ordinate system and orthographic imaging conditions are assumed. We will denote the point in space by (X,Y,Z), the two basis view co-ordinates by (x',y') and (x'',y'') respectively, and our target view co-ordinates by (x,y). We choose our world co-ordinate axes to be in the same orientation as the co-ordinate system defined by the camera for the first basis view so that the projection equations for this basis view are those given in chapter 2, equation (2.3.2):

$$\begin{aligned} x' &= X \\ y' &= Y \end{aligned} \qquad (3.3.1)$$

We let the camera for the second basis view be positioned by rotating the camera used for the first basis view about the *Y*-axis by an angle ϕ, such that the projection equations are:

$$\begin{aligned} x'' &= X\cos\phi + Z\sin\phi \\ y'' &= Y \end{aligned} \qquad (3.3.2)$$

Similarly we let the camera used for the target view be positioned by rotating the camera used for the first basis view about the Y-axis by an angle θ. Thus the target view projection equations are:

$$\begin{aligned} x &= X\cos\theta + Z\sin\theta \\ y &= Y \end{aligned} \qquad (3.3.3)$$

If we eliminate the X, Y and Z from equations (3.3.1), (3.3.2) and (3.3.3) we obtain the pair of multi-view relationships:

$$\begin{aligned} x &= ax' + bx'' \\ y &= y' = y'' \end{aligned} \qquad (3.3.4)$$

where a and b are constants given by, $a = \dfrac{\sin(\phi - \theta)}{\sin\phi}$ and $b = \dfrac{\sin\theta}{\sin\phi}$.

By giving this simplified example we have shown that, in a restricted case, it is possible to represent the target view as a combination of the two basis views. Ullman and Basri [Bas93, UB91, Ull96] go on to show that, on the assumption that all images are obtained under orthographic projection, the linear combination property generalises to any 3D transformation of the cameras. The same result applies for the general affine camera. This can be seen from the general expressions for the affine projection given by equation (2.3.10) in chapter 2, where each image co-ordinate x and y can be written as a linear combination of the world co-ordinates, X, Y and Z:

$$\begin{aligned} x &= A_1X + A_2Y + A_3Z + A_4 \\ y &= B_1X + B_2Y + B_3Z + B_4 \end{aligned} \qquad (3.3.5)$$

where the A_i, B_i are constants. Two basis views provides us with four equations:

$$\begin{aligned} x' &= A_1'X + A_2'Y + A_3'Z + A_4' \\ y' &= B_1'X + B_2'Y + B_3'Z + B_4' \\ x'' &= A_1''X + A_2''Y + A_3''Z + A_4'' \\ y'' &= B_1''X + B_2''Y + B_3''Z + B_4'' \end{aligned} \qquad (3.3.6)$$

which is sufficient, in principle, to solve for the world co-ordinates X, Y and Z. In fact, since there are only three variables, only three of the four equations in (3.3.6) ("1½ views") are needed to solve for the co-ordinates. These expressions for X, Y and Z can then be substituted into the equations for other affine views, for example (3.3.5), to obtain a pair of relationships between three affine views, such as:

$$x = a_1 x' + a_2 y' + a_3 x'' + a_4$$
$$y = b_1 x' + b_2 y' + b_3 x'' + b_4 \qquad (3.3.7)$$

The equations in (3.3.7) are the same as those derived by Basri in [Bas93]. We notice that there is no y'' term in (3.3.7), this is because, as stated above, only three of the equations in (3.3.6) are needed to solve for the world co-ordinates, and we have chosen to omit the equation in y''. This is not the only possible choice, we could have chosen to use any three of the four equations. Consequently (3.3.7) is not the only possible combination of the two basis views that can be used to represent x and y.

In practice, when estimating the relationships from point correspondences, the locations of the points will be subject to measurement errors and the equations in (3.3.7) will not be exact. There will also be errors in the relationships that arise when the affine imaging conditions are used as an approximation to perspective. It is possible to use an overcomplete set of equations that are symmetric in the two basis views:

$$x = a_1 x' + a_2 y' + a_3 x'' + a_4 y'' + a_5$$
$$y = b_1 x' + b_2 y' + b_3 x'' + b_4 y'' + b_5 \qquad (3.3.8)$$

Equations (3.3.8) were given in [BSG98, KB98, KBG99] have been used previously in [Han99, HB00a, HB00b]. We give them here as they will be used in chapter 4 as the starting point for the discussion on encoding of views.

If we return to (3.3.7) we note that it is possible to simplify this pair of equations a little by using centre-of-mass co-ordinates. In this case each image point (x, y) is replaced by $(\Delta x, \Delta y)$, defined by:

$$\Delta x_i = x_i - \bar{x}$$
$$\Delta y_i = y_i - \bar{y} \qquad (3.3.9)$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and n is the number of points. Centre-of-mass co-ordinates are also known as registered image co-ordinates [BL98, TK92], relative co-ordinates or difference vectors [SZB95, TM99]. They are often used because they remove effects of translation in the image plane [SZB95]. They also lead to the definition of a registered camera [TM99] or centred affine camera [BL98] by setting $q_{14} = q_{24} = 0$ in the affine camera matrix given in (2.3.9):

$$M_{registered} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & 0 \\ q_{21} & q_{22} & q_{23} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} . \tag{3.3.10}$$

When the registered camera is used, the projection equations become:

$$\begin{aligned} x &= q_{11}X + q_{12}Y + q_{13}Z \\ y &= q_{21}X + q_{22}Y + q_{23}Z \end{aligned} . \tag{3.3.11}$$

If we now consider the relationships between points in three views that have been obtained using registered cameras, we find that the relationships (3.3.7) become:

$$\begin{aligned} x &= a_1x' + a_2y' + a_3x'' \\ y &= b_1x' + b_2y' + b_3x'' \end{aligned} . \tag{3.3.12}$$

The pair of equations given in (3.3.12) are the same as those derived by Ullman and Basri in [UB91] and [Ull96]. Similarly we can write equation (3.3.8) as:

$$\begin{aligned} x &= a_1x' + a_2y' + a_3x'' + a_4y'' \\ y &= b_1x' + b_2y' + b_3x'' + b_4y'' \end{aligned} , \tag{3.3.13}$$

when each of the co-ordinates, (x, y), $(x', y')$and (x'', y'') are expressed using centre of mass co-ordinates. Equations (3.3.13) are the form most recently used, for example, by Dias and Buxton [DB02] in an extension of the technique to flexible objects.

3.3.2 Affine Multi-View Tensors

In the previous section we have shown that, in the case of the affine camera, the relationships between three views can be determined by elimination of the world co-ordinates, X, Y and Z, from the projection equations for the three views. It is also possible to express the relationships using tensor notation, similar to that used in the perspective case to define the trifocal tensor (3.2.4) and quadrifocal tensor (3.2.7).

Mendonça and Cipolla defined an affine trifocal tensor, [MC98], by using the formula for the perspective trifocal tensor given in (3.2.4). In the perspective case, this formula assumes one of the projection matrices to be of a special form. Likewise, in the affine case, the world co-ordinate system is chosen such that the one of the projection matrices is of the form:

$$M_{aff} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} . \tag{3.3.14}$$

The second and third camera matrices, M' and M'', are general affine camera matrices defined by:

$$M'_{aff} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & a_{34} \end{pmatrix}, \text{ and } M''_{aff} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ 0 & 0 & 0 & b_{34} \end{pmatrix}. \qquad (3.3.15)$$

The trifocal tensor is then defined, as in the perspective case, by using the formula:

$$T_{aff\,ijk} = M'_{aff\,ji} M''_{aff\,k4} - M'_{aff\,j4} M''_{aff\,ki} \quad , \qquad (3.3.16)$$

where $i, j, k = 1,..,3$. Whereas the perspective trifocal tensor has 27 elements, in the affine case 11 of these will be equal to zero as shown in [MC98]. The affine trifocal tensor thus has 16 non-zero elements, the overall scale of which is unimportant. It is known that the affine trifocal tensor has 12 degrees of freedom [MC98] (there are 8 for each of the three camera matrices minus 12 for a 3D affine transformation), so there are 3 constraints on the elements of T_{aff}. This means that the affine trifocal tensor is simpler than the perspective trifocal tensor, not only by having fewer elements, but also having fewer constraints on them.

In the previous section we discussed how the relationships can be simplified by registering the views. Similarly it is possible to define a centred affine trifocal tensor [BL98, DB02] or registered trifocal tensor [HTM99, TM99]. The registered trifocal tensor has only 12 non-zero parameters and only 9 degrees of freedom [HTM99]. The registered or centred affine trifocal tensor (CATT) was used by Dias and Buxton [DB02], who give details of the constraints.

We mentioned in section 3.2.2 that it is possible to form a common framework for the perspective multiple-view geometry and use it to define the multi-view relationships. A similar framework has been used in the affine case [TK92, TM99] to explore the affine multi-view relationships. This has been done for both the registered and unregistered cases [HTM99, TM99]. By using this common framework an affine quadrifocal tensor has also been defined in [HTM99]. The definition for the affine quadrifocal tensor is the same as in the perspective case:

$$Q_{aff\,pqrs} = \det \begin{bmatrix} M_{aff\,p} \\ M'_{aff\,q} \\ M''_{aff\,r} \\ M'''_{aff\,s} \end{bmatrix} \quad p,q,r,s = 1..3 \quad , \qquad (3.3.17)$$

where $M_{aff\,a}$ corresponds to the a^{th} row of the affine projection matrix. In the unregistered case the affine quadrifocal tensor has 47 non-zero elements and 20 degrees of freedom whereas in the registered case there are 31 non-zero elements and 15 degrees of freedom [HTM99, TM99].

As in the perspective case, the affine quadrifocal tensor provides no further information about the scene structure than the affine trifocal tensor. However, using the common framework for multiple images allows us to find an expression for the quadrifocal tensor in terms of the trifocal tensors. Therefore it is possible to use the affine quadrifocal tensor to provide a consistent set of affine trifocal tensors between four views [HTM99].

3.4 Encoding of Views

So far in this chapter we have introduced the different multi-view relationships and have discussed how they can be expressed in terms of the entries in the projection matrix. Since we usually work with uncalibrated cameras, the projection matrices will not be known and the multi-view relationships will need to be estimated directly from the images. Section 3.4.1 discusses how the multi-view relationships can be estimated from a set of corresponding control points in the images. As mentioned in the introduction, how we find the correspondences between the images is not part of the work covered in this thesis and we will assume that they are already known.

The multi-view relationships provide a mapping between the corresponding control points in the views but we also need a method of encoding the image intensities. In section 3.4.2 we consider how the intensities in the target view may be represented as a function of the intensities in the two basis views.

3.4.1 Estimating the Multi-View Relationships

In this section we introduce the methods that can be used to estimate the multi-view relationships from a set of corresponding points in the images. We pay particular attention to the linear methods, where the relationships are found by solving a set of equations that are linear in the unknown parameters, for example linear in the elements of F or T. These methods work well for the affine relationships because

they are linear in the image co-ordinates and because there are few constraints. Neglecting the constraints is therefore less serious than it would be for the perspective case when there are more of them. Not enforcing the constraints when using the perspective relationships can lead to inaccurate results. This is unlike the affine case, where we can neglect the constraints and still achieve accurate results.

There are non-linear algorithms that can be used to give a more accurate estimation of the perspective relationships, or to include the constraints in either the perspective or affine cases. An example of the latter is given in Dias and Buxton [DB02]. Although these non-linear algorithms can give a more accurate result than the linear algorithms they are considerably more complicated that the linear methods [LDFP93, LF96, TZ97] and, if not used carefully, can be very sensitive to errors. However, it has been shown that using a simple transformation of the points leads to improvements in the results of the linear estimation of the fundamental matrix [Har97b, Har95c]. It was later shown that it is possible to improve the results further by careful consideration of where the errors occur [MM98].

We will begin by looking at the pair of affine three-view relationships given in equation (3.3.8) and show how they can be solved using a simple linear least squares technique [GV96, LH95]. We choose this pair as they will be used for evaluation purposes in the next chapter, but the principles of the solution can be applied to any of the affine relationships given in section 3.3.1. Suppose that we have a set of n corresponding control points in the three images. Then for each control point i we can write a pair of relationships of the form given in (3.3.8):

$$\begin{aligned} x_i &= a_1 x_i' + a_2 y_i' + a_3 x_i'' + a_4 y_i'' + a_5 \\ y_i &= b_1 x_i' + b_2 y_i' + b_3 x_i'' + b_4 y_i'' + b_5 \end{aligned} \qquad (3.4.1)$$

In the case where $n = 5$ we have ten equations (two equations for each correspondence) in the ten unknowns, a_1 to a_5 and b_1 to b_5, and we are able to solve for the relationships. In the case where $n > 5$ we may use a least squares technique [LH95, GV96] to obtain a solution. Since there will inevitably be errors associated with the locations of the control points, equations (3.4.1) will only be approximately satisfied and we may include error terms in the usual way and say:

$$\begin{aligned} x_i + \varepsilon_i &= a_1 x_i' + a_2 y_i' + a_3 x_i'' + a_4 y_i'' + a_5 \\ y_i + \eta_i &= b_1 x_i' + b_2 y_i' + b_3 x_i'' + b_4 y_i'' + b_5 \end{aligned} \qquad (3.4.2)$$

It is then possible to find a solution for the a_i and b_i such that the sum of the squared errors is minimised. Firstly we re-write equations (3.4.2) using a matrix representation:

$$\begin{aligned} \underline{x}+\underline{\varepsilon} &= D\underline{a} \\ \underline{y}+\underline{\eta} &= D\underline{b} \end{aligned} \quad , \tag{3.4.3}$$

where $\underline{x}$ and $\underline{y}$ are the co-ordinate vectors, $\underline{\varepsilon}$ and $\underline{\eta}$ are the error vectors, D is the $n \times 5$ design matrix containing the co-ordinates of points in the basis views and $\underline{a}$ and $\underline{b}$ are vectors of the unknown parameters. Thus:

$$\begin{aligned} x &= (x_1 \quad x_2 \quad . \quad . \quad x_n)^T & \underline{y} &= (y_1 \quad y_2 \quad . \quad . \quad y_n)^T \\ \underline{\varepsilon} &= (\varepsilon_1 \quad \varepsilon_2 \quad . \quad . \quad \varepsilon_n)^T & \underline{\eta} &= (\eta_1 \quad \eta_2 \quad . \quad . \quad \eta_n)^T \\ \underline{a} &= (a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5)^T & \underline{b} &= (b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5)^T \end{aligned} \quad ,$$

and $$D = \begin{pmatrix} x_1' & y_1' & x_1'' & y_1'' & 1 \\ x_2' & y_2' & x_2'' & y_2'' & 1 \\ . & . & . & . & . \\ . & . & . & . & . \\ x_n' & y_n' & x_n'' & y_n'' & 1 \end{pmatrix} \quad . \tag{3.4.4}$$

We wish to find solutions for the vectors $\underline{a}$ and $\underline{b}$ such that the total sum of squared errors, $\|\underline{\varepsilon}\|^2 + \|\underline{\eta}\|^2$, is minimised. We can re-arrange equations (3.4.3) to obtain expressions for $\underline{\varepsilon}$ and $\underline{\eta}$. We can then differentiate with respect to $\underline{a}$ and $\underline{b}$ and set the derivatives equal to zero to obtain the following equations:

$$\begin{aligned} \underline{a} &= (D^T D)^{-1} D^T \underline{x} \\ \underline{b} &= (D^T D)^{-1} D^T \underline{y} \end{aligned} \quad . \tag{3.4.5}$$

Equations (3.4.5) are known as the normal equations [GV96, LH95, VV91]. In practice they are never computed directly as they exacerbate any ill-conditioning in the matrices. Instead they are solved using a more stable method, for example, by using Householder or Givens rotations [GV96] or by using Cholesky decomposition [LH95]. Alternatively, as we choose to do, they may be solved by using a singular value decomposition [LH95, GV96, VV91] of the design matrix D. In cases where the matrix is singular (or close to singular) the singular value decomposition method is particularly useful as it allows explicit conditioning of the solution [LH95, VV91]. Details of singular value decomposition and how it can be used to solve least squares

problems are given in appendix A. It is also possible to find solutions for $\underline{a}$ and $\underline{b}$ by computing the pseudo-inverse of the matrix D as stated in appendix A.

Equations (3.4.3) would be appropriate if the only errors were on the (x_i, y_i) control points. Since this is unlikely to be the case a solution based on equations (3.4.5) in practice will not distribute the errors correctly and is likely to lead to estimates of the parameters that are not as accurate as they could be. It may be better, therefore, to use a total least squares solution (as will be described in section 4.2.1) to solve for the parameters.

In the above problem we had a set of linear inhomogeneous equations (3.4.3) which we said may be solved by using singular value decomposition. SVD may also be used to find linear solutions to the perspective multi-view relationships and to find the affine fundamental matrix [HZ00]. To do so we write the multi-view relationships as a system of homogeneous linear equations. For example, given $n \geq 5$ corresponding points in the two views, we are able to write down a system of equations for the elements $a...e$ of the affine fundamental matrix (2.5.3):

$$\underbrace{\begin{pmatrix} x_1 & y_1 & x_1' & y_1' & 1 \\ x_2 & y_2 & x_2' & y_2' & 1 \\ . & . & . & . & . \\ . & . & . & . & . \\ x_n & y_n & x_n' & y_n' & 1 \end{pmatrix}}_{D} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ 0 \end{pmatrix} \quad . \tag{3.4.6}$$

We can write down a similar system of equations for the perspective fundamental matrix. Recall from (2.2.10) the equation relating a corresponding point in two images:

$$p'^T F p = 0 \quad , \tag{3.4.7}$$

where F is the 3×3 matrix $\begin{pmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{pmatrix}$. Given a set of n corresponding points in the two images we are able to write down a system of equations:

$$A\underline{f} = 0 \quad , \tag{3.4.8}$$

where A is the design matrix and $\underline{f} = (f_1 \quad f_2 \quad . \quad . \quad f_9)^T$. Each point correspondence contributes one row to the design matrix:

$$A=\begin{pmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2x_2 & x'_2y_2 & x'_2 & y'_2x_2 & y'_2y_2 & y'_2 & x_2 & y_2 & 1 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{pmatrix} \quad . \qquad (3.4.9)$$

In both the affine and perspective cases the design matrix will not be exact, since there will be measurement errors on the location of the control points. This means that the right-hand side in (3.4.6) or (3.4.8) will not be equal to zero, but to an error vector. We seek to find a least squares solution such that the total sum of squared errors is minimised. For example, we want to find a solution for $\underline{f}$ such that the value of $\|A\underline{f}\|^2$ is minimised. In order to find a unique solution we impose the constraint $\|\underline{f}\|=1$. A solution for the vector $\underline{f}$ may then be found by using SVD as outlined in appendix A [TV98].

We have said that it is possible to find a solution for the fundamental matrix by using SVD to solve a homogenous system of linear equations. The same principles can be used to estimate other multi-view relationships. For example, Hartley uses a linear method for estimating the trifocal tensor in [Har97a].

It was mentioned earlier that more sophisticated non-linear methods exist for estimating the perspective relationships. Iterative methods for estimating the fundamental matrix can be found in [LDFP93, LF96] and a robust method of computing the trifocal tensor is given by Torr and Zisserman [TZ97, TZ96]. The advantage of using the linear methods is their simplicity. The disadvantages are that the appropriate constraints are not taken into account and that they produce less accurate results.

In the case of the fundamental matrix there is only one constraint on its elements, that is $\det F=0$. Given a solution for F we are able to enforce the constraint by finding the SVD of F and setting the smallest singular value equal to zero [Har97a, TV98]. In addition Hartley showed that it is possible to improve the accuracy of the linear method by normalising the points. We do not give the details here but they can be found in [Har97b, Har95c]. It has also been shown that the results can be improved further by careful consideration of where the errors occur [MM98] and using a total least squares solution [GV80, VV91]. The total least squares solution will be discussed in detail in the next chapter.

3.4.2 Estimation of the Target View Intensities

We have explored the multi-view relationships and shown how they can be linearly estimated from a set of corresponding control points in the images. If we wish to encode a view using a pair (or more) of basis views then we need a method of expressing the target view intensities in terms of those in the basis views. The difficulty is that the appearance of an object is affected by the lighting conditions and the reflectance properties of surfaces in the scene [WP98, WW92]. If the surface is a perfect specular reflector the light is reflected from the surface at an angle equal to the angle of incidence, whereas light reflected off a perfect diffuse surface is scattered equally in all directions. In most cases real surfaces are neither perfect specular or perfect diffuse reflectors. They have both specular and diffuse components [WP98].

It is also necessary to consider the possibility that the illumination conditions may not be constant across a sample of images. If, for example, the images are obtained on separate occasions under varying conditions, for example outdoors at different times of the day, then there is the additional difficulty of the varying illumination. This problem of varying illumination is known as the photometric problem [Sha92, TCD01, YM98]. Under these circumstances it may be beneficial to use an accurate model of the reflectance properties of surfaces in the scene of interest [Zha98], such as the Bidirectional Reflection Distribution Function (or BRDF) [HS79, KV96, WP98, WW92]. The BRDF characterises the reflectance of a surface as a function of the illumination conditions and the viewing position.

If we assume that the surfaces are diffuse reflectors within the viewpoint range considered and that the images are of a static scene with fixed illumination, then it is possible to interpolate the basis images in order to render the target view. To give an example of this in the affine case we return to the pair of affine (three-view) relationships given in equation (3.3.8) and outline the method of rendering the target view intensities used in [BSG98, HB00a, KB99].

Recall from (3.3.8):

$$\begin{aligned} x &= a_1 x' + a_2 y' + a_3 x'' + a_4 y'' + a_5 \\ y &= b_1 x' + b_2 y' + b_3 x'' + b_4 y'' + b_5 \end{aligned} \qquad (3.4.10)$$

We would now like to use the intensities in the two basis views, denoted by I' and I'', to estimate the intensity in the target view, I. In order to do so it is useful to obtain a relative measure of how similar (close) the target view is to each of the basis views. Such distances were defined in [BSG98 and KB99] using the co-efficients of the relationships in (3.4.10):

$$\begin{aligned} d'^2 &= a_3^2 + a_4^2 + b_3^2 + b_4^2 \\ d''^2 &= a_1^2 + a_2^2 + b_1^2 + b_2^2 \end{aligned} \quad , \tag{3.4.11}$$

where d' and d'' are the relative distances from the first and second basis views respectively.

Given these distances the target view intensities can be set using an interpolation of the basis view intensities according to the equation [BSG98]:

$$I = w'I' + w''I'' \quad , \tag{3.4.12}$$

where $w' = \dfrac{d''^2}{d'^2 + d''^2}$ and $w'' = \dfrac{d'^2}{d'^2 + d''^2}$.

This scheme describes how to render the intensities at the control points (x_i, y_i) for which we know the corresponding points in the two basis views, (x_i', y_i') and (x_i'', y_i''). We actually need to apply equation (3.4.10) to every pixel in the target view (x, y) for which the corresponding points, (x', y') and (x'', y''), in the two basis views are unknown. In order to render the intensities at every pixel in the target view we begin by triangulating the target image using the control points and then use a linear piecewise mapping function inside each triangle as described in [Gos86] and used previously in [KB99, KBG99, HB00a, and HB00b]. When we talk about triangulating the image we mean dividing the image into triangular regions in order to render the intensities, not triangulating the scene as part of a reconstruction algorithm.

We will now show that using the piecewise linear mapping to determine the dense correspondence across the images is consistent with using the affine multi-view relationships to synthesise views. Given a set of corresponding points (x_i, y_i), (x_i', y_i'), (x_i'', y_i'') $i = 1...n$, let us represent a point (x, y) in the novel view I as:

$$x = \sum_i \lambda_i x_i \quad \text{and} \quad y = \sum_i \lambda_i y_i \quad , \tag{3.4.13}$$

where $\sum_i \lambda_i = 1$. We can then estimate the corresponding points in the two basis views I' and I'' as:

$$x' = \sum_i \lambda_i x_i' \qquad y' = \sum_i \lambda_i y_i'$$
$$x'' = \sum_i \lambda_i x_i'' \quad , \qquad y'' = \sum_i \lambda_i y_i'' \quad . \tag{3.4.14}$$

Although this does not guarantee the correspondence of (x, y), (x', y') and (x'', y'') is correct, thanks to the linearity it is consistent with the affine view relationships. In particular, if the control points satisfy equation (3.4.10), then so do (x, y), (x', y') and (x'', y''). For example, in detail:

$$\begin{aligned} a_1 x' + a_2 y' + a_3 x'' + a_4 y'' + a_5 &= \sum_i \lambda_i \left(a_1 x_i' + a_2 y_i' + a_3 x_i'' + a_4 y_i'' + a_5 \right) \\ &= \sum_i \lambda_i x_i \qquad , \qquad (3.4.15) \\ &= x \end{aligned}$$

as required.

3.5 Synthesis of Novel Views

We have seen how the multi-view relationships can be estimated from a set of control points and how they can be used to encode an existing view. In order to use the multi-view relationships to synthesise a new view it is necessary to estimate either the positions of the control points in the target view or the multi-view relationships that will be used to synthesis the novel view. We may use interpolation techniques to estimate the positions of a set of control points in a target image that is "in-between" the basis views. Seitz and Dyer synthesised novel views by interpolating between a pair of basis views [SD95]. They showed that if the basis view cameras are parallel, i.e., the optical axes of the two cameras are parallel and the cameras lie in a common plane orthogonal to the optical axes, then interpolating between these views produces realistic looking synthesised views. If the basis views are not parallel then it is necessary to first rectify the images in order for their method to produce realistic results [SD96, SD95].

Pollard et al have also used an interpolation technique to synthesise novel views. However, in their method they do not work with rectified images. Instead of using points they match edges between a set of three basis views and interpolate between the triplet of basis views to obtain the locations of the edges in the novel view [PH99, PHPL97, PPHL98].

The alternative to interpolating points and edges in the images is to estimate the multi-view relationships between known basis views and a novel target view. This can be done by finding a suitable parameterisation of the multi-view relationships. An example of parameterising the relationships is given in [HB00a] where the multi-view relationships used are the affine relationships given in (3.3.8). Hansard and Buxton constrain the movement of the camera to rotate around a fixation point in a horizontal plane. They then parameterise the co-efficients in terms of the angle of rotation.

Avidan and Shashua have also synthesised novel views by estimating the multi-view relationships rather than interpolating features in the image [AS98, AS97]. They use the trifocal tensor between three known views to estimate the tensor between two of these views and a novel view. They do this by expressing the novel tensor in terms of the original tensor and the camera motion parameters (rotation and translation) between one of the known views and the novel view [AS98].

One of the advantages of synthesising novel views by estimating the new multi-view relationships rather than interpolating the basis images is that it is possible to extrapolate from the basis views [AS98]. This enables a wider range of views to be synthesised. In so doing, it would be very useful to know how far we can extrapolate from the basis views and still achieve a realistic novel view. In chapter 6 we discuss how the structure within the total least squares solution can be used, under certain viewing conditions, to obtain an estimate of when the view synthesis procedure, described in the following two chapters, breaks down.

In chapter 5 we give a method of parameterising a set of sample images. This parameterisation does not constrain the movement of the camera in any way and can be used to parameterise either the positions of the control points or the co-efficients of the multi-view relationships. The method imposes the constraint that the control points or the co-efficients of the multi-view relationships should vary as smoothly as possible as the parameters vary.

Chapter 4

Encoding of Views and the Total Least Squares Solution

4.1 Overview

In this chapter we introduce the total least squares solution and discuss how it can be used to encode views of an object/scene as a combination of other views. We begin in section 4.2 by forming a new pair of relationships between three views. These three-view relationships are based on the affine imaging assumptions. We show that by carefully considering where the errors occur, these relationships can be estimated from a set of control points in the three images by using a total least squares solution. In section 4.3 we evaluate the new affine three-view relationships by comparing them with the affine relationships given by equation (3.3.8) in chapter 3. In both cases the relationships are estimated from the same set of corresponding control points in the three images. We then compare the total sum of squared errors obtained in the two different cases.

We then move on, in section 4.4, to how these relationships can be used for the encoding of views. We discuss how the control points can be used to estimate dense correspondence between the images. A method of setting the target view intensities is also given in section 4.4. In section 4.5 we implement and evaluate the method of encoding views of an object/scene that we have introduced in this chapter.

When reconstructing a view we use the assumption that the surfaces in the image are made up of planar patches. In section 4.6 we evaluate how this assumption affects the reconstructed intensities when the surfaces are not planar. Finally, in section 4.7, we discuss how the total least squares method of encoding views can be extended to include more than two basis views.

4.2 Linear Combination of Views and the Total Least Squares Solution

We have seen in chapter 3, section 3.3.1, that by assuming affine imaging conditions it is possible to represent an image as a linear combination of 2 views by choosing one of four possible combinations of image co-ordinates [Bas93]:

$$\begin{aligned} x &= a_1 x' + a_2 y' + a_3 x'' + a_4 \\ y &= b_1 x' + b_2 y' + b_3 x'' + b_4 \end{aligned} \qquad (4.2.1)$$

We mentioned in section 3.3.1 that this is not the only possible pair of such relationships between three affine views. The target view co-ordinates x and y can be expressed as a linear combination of any three of the four basis view co-ordinates, x', y', x'' and y''. In the absence of errors, equations (4.2.1) will be exact in the case of affine cameras. In practice, control points are likely to contain measurement errors and it will not be possible to determine the relationships in equation (4.2.1) exactly. We mentioned earlier in section 3.3.1 that it is possible to use an overcomplete set of equations [BSG98]:

$$\begin{aligned} x &= a_1 x' + a_2 y' + a_3 x'' + a_4 y'' + a_5 \\ y &= b_1 x' + b_2 y' + b_3 x'' + b_4 y'' + b_5 \end{aligned} \qquad (4.2.2)$$

The pair of equations in (4.2.2) have the advantage of being symmetrical in the two basis views. This is useful for interpolation between a pair of basis views as it gives the correct analytic form close to either. For example, if the target view, (x, y), is close to the second basis view, (x'', y''), then using the relationships given in (4.2.1) will lead to large errors in the relationships as we do not include the term y''. Under these circumstances it is more appropriate to use the relationships in equation (4.2.2).

Given a set of control points it is possible to solve for the co-efficients, a_i and b_i, by using a least squares solution [GV96, LH95] as described in section A.1 of appendix A. However, this solution assumes that all the measurement errors are contained within the target view co-ordinates. In practice this will not be the case. The locations of the basis view control points will inevitably have errors whether they are selected by hand or automatically. There will also be errors on the target view co-ordinates. In the case where we are encoding existing views the target view and basis view control points are usually all located using the same techniques. This means that the errors on the control points in each of the views will be similar, i.e., we can

assume that the errors on the control points are independent and identically distributed. If we make this assumption the appropriate method for estimating the relationships is the total least squares solution [GV80, VV91]. The use of the total least squares solution to determine the multi-view relationships will be discussed in the next section.

If we are not encoding existing views, but instead wish to use the combination of views to synthesise novel views, then the location of the control points are found by using interpolation/extrapolation of the basis view control points. In this case there will still be errors on the target view co-ordinates but these will be propagated from the errors on the basis view control points. Under these circumstances, where there may be correlation between the errors on the target view co-ordinates and the basis view co-ordinates, it is possible to solve for the co-efficients by using a generalised total least squares solution [VV89]. A brief explanation of the generalised total least squares problem will be given in section 4.2.2.

4.2.1 Application of the Total Least Squares Problem

We have said that the total least squares solution can be used to estimate the multi-view relationships when all terms in the relationships are equally likely to contain the same (or similar) measurement errors. The pair of equations (4.2.2) are overcomplete and symmetrical in the basis views, but the target view co-ordinates are treated differently from the basis view co-ordinates. In order to keep everything symmetrical between the target view and the basis views we choose to seek a pair of linear relationships between the three views of the form:

$$\begin{aligned} l_1x + l_2y + l_3x' + l_4y' + l_5x'' + l_6y'' + l_7 &= 0 \\ m_1x + m_2y + m_3x' + m_4y' + m_5x'' + m_6y'' + m_7 &= 0 \end{aligned} \qquad (4.2.3)$$

The pair of relationships in equation (4.2.3) are symmetrical in all the co-ordinates. Given a set of seven corresponding points then we are able to solve for the co-efficients l_i and m_i. In practice we may have more than seven corresponding control points in the three images. In this case we can not satisfy the relationships exactly. Instead for each control point, i there will be some error on the relationships and we can include this error term in the relationships:

$$\begin{aligned} l_1x_i + l_2y_i + l_3x_i' + l_4y_i' + l_5x_i'' + l_6y_i'' + l_7 &= \varepsilon_i \\ m_1x_i + m_2y_i + m_3x_i' + m_4y_i' + m_5x_i'' + m_6y_i'' + m_7 &= \eta_i \end{aligned} \quad . \qquad (4.2.4)$$

We wish to find a solution that minimises the error on the relationships. We can solve for the co-efficients l_i and m_i in equation (4.2.4) under the assumption that the errors are independently and identically distributed among all the co-ordinates in all three views. It is possible to rewrite equations (4.2.4) using matrix notation:

$$\begin{aligned} D\underline{l} &= \underline{\varepsilon} \\ D\underline{m} &= \underline{\eta} \end{aligned} \quad , \qquad (4.2.5)$$

where $D = \begin{pmatrix} x_1 & y_1 & x_1' & y_1' & x_1'' & y_1'' & 1 \\ x_2 & y_2 & x_2' & y_2' & x_2'' & y_2'' & 1 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ x_n & y_n & x_n' & y_n' & x_n'' & y_n'' & 1 \end{pmatrix}$ is the design matrix, $\underline{l} = (l_1, l_2, ..., l_7)^T$,

$\underline{m} = (m_1, m_2, ..., m_7)^T$, $\underline{\varepsilon} = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_n)^T$ and $\underline{\eta} = (\eta_1, \eta_2, ..., \eta_n)^T$.

In the case where every entry in the matrix D is equally likely to contain measurement errors it is possible to solve for the co-efficient vector $\underline{l}$ and $\underline{m}$ using a total least squares technique. The design matrix we have here contains a column of 1's. The appropriate technique for this type of problem is the mixed least squares-total least squares (LS-TLS) solution [VV91]. The mixed LS-TLS problem arises when the design matrix contains columns that are known exactly (i.e. there are columns that are error free) and the errors in the remaining columns are assumed to be independently and identically distributed [VV91].

The solution to the mixed LS-TLS problem maybe found in general by using a QR decomposition of the design matrix D. Full details of the mixed LS-TLS problem/solution are described in appendix B. In our particular case, where only one of the columns in the design matrix is error free, it is possible to transform the problem into a classical total least squares problem by eliminating the constant terms l_7 and m_7 from equations (4.2.4).

By summing the relationships in equation (4.2.4) over all the control points we are able to obtain expressions for l_7 and m_7 in terms of the control points and the remaining co-efficients. If we sum the relationships (4.2.4) over all the control points we obtain:

$$\begin{aligned}\sum_{i=1}^{n}\left(l_1x_i+l_2y_i+l_3x_i'+l_4y_i'+l_5x_i''+l_6y_i''+l_7\right)&=\sum_{i=1}^{n}\varepsilon_i\\ \sum_{i=1}^{n}\left(m_1x_i+m_2y_i+m_3x_i'+m_4y_i'+m_5x_i''+m_6y_i''+m_7\right)&=\sum_{i=1}^{n}\eta_i\end{aligned} \quad . \qquad (4.2.6)$$

Since we have made the assumption that the errors are independent and identically distributed we can assume that as n tends to infinity the sum of the error terms tends to zero, i.e., that the solution is unbiased in the limit. By assuming zero mean independent and identically distributed errors we obtain:

$$\sum_{i=1}^{n}\varepsilon_i \approx \sum_{i=1}^{n}\eta_i \approx 0 \quad . \qquad (4.2.7)$$

By using this fact we are able to re-arrange (4.2.6) to obtain expressions for l_7 and m_7:

$$\begin{aligned}l_7&=\frac{1}{n}\sum_{i=1}^{n}\left(l_1x_i+l_2y_i+l_3x_i'+l_4y_i'+l_5x_i''+l_6y_i''\right)\\ m_7&=\frac{1}{n}\sum_{i=1}^{n}\left(m_1x_i+m_2y_i+m_3x_i'+m_4y_i'+m_5x_i''+m_6y_i''\right)\end{aligned} \quad . \qquad (4.2.8)$$

As shown in appendix B, these expressions for l_7 and m_7 are equivalent to those obtained when solving the mixed LS-TLS problem by QR decomposition of the design matrix D. We can now substitute these expressions into equations (4.2.4) to obtain:

$$\begin{aligned}l_1\Delta x_i+l_2\Delta y_i+l_3\Delta x_i'+l_4\Delta y_i'+l_5\Delta x_i''+l_6\Delta y_i''&=\varepsilon_i\\ m_1\Delta x_i+m_2\Delta y_i+m_3\Delta x_i'+m_4\Delta y_i'+m_5\Delta x_i''+m_6\Delta y_i''&=\eta_i\end{aligned} \quad . \qquad (4.2.9)$$

Where $\Delta x_i = x_i - \frac{1}{n}\sum_{i=1}^{n}x_i$. Equations (4.2.9) provide a pair of relationships between the three views when the control points are expressed using centre of mass co-ordinates as defined in section 3.3.1 [BL98, TK92]. Here we have seen that it is possible to eliminate the constant terms in equations (4.2.4) by expressing the image points using centre of mass co-ordinates and assuming zero-mean independent and identically distributed errors. This is consistent with the previous work on registered views [SZB95, TM99] that we discussed in section 3.3.1.

We can re-write our system of equations (4.2.9) as:

$$\begin{aligned}D\underline{l}&=\underline{\varepsilon}\\ D\underline{m}&=\underline{\eta}\end{aligned} \quad , \qquad (4.2.10)$$

where now we let $D = \begin{pmatrix} \Delta x_1 & \Delta y_1 & \Delta x_1' & \Delta y_1' & \Delta x_1'' & \Delta y_1'' \\ \Delta x_2 & \Delta y_2 & \Delta x_2' & \Delta y_2' & \Delta x_2'' & \Delta y_2'' \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \Delta x_n & \Delta y_n & \Delta x_n' & \Delta y_n' & \Delta x_n'' & \Delta y_n'' \end{pmatrix}$, and $\underline{l} = (l_1, l_2, ..., l_6)^T$,

$\underline{m} = (m_1, m_2, ..., m_6)^T$, $\underline{\varepsilon} = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_n)^T$ and $\underline{\eta} = (\eta_1, \eta_2, ..., \eta_n)^T$.

By using centre of mass co-ordinates we have transformed our design matrix such that all the entries are equally likely to contain the same amount of error. This simple transformation of the image points has transformed our problem from a mixed LS-TLS problem into a classical total least squares (TLS) problem [GV96, LH95, VV91]. The solution to the total least squares problem that minimises the total sum of the squared errors, $\|\varepsilon_i\|^2 + \|\eta_i\|^2$, can be obtained by a singular value decomposition (see appendix A) of the design matrix D. The solutions for $\underline{l}$ and $\underline{m}$ are the singular vectors corresponding to the two smallest singular values of D.

When we perform the SVD of the matrix D we find that three of the singular values, w_i are small compared to the other three. This implies that there are potentially three independent relationships between the co-ordinates of the three views. This is what we would expect, as we have shown earlier section in 3.1.3: when there are no observation errors and the cameras are all affine, there are three independent relationships between three views. We only require two of these relationships in order to recover the target view co-ordinates. We choose the two singular vectors corresponding to the two smallest singular values, as they will provide the best relationships (lowest errors) between the image co-ordinates.

4.2.2 The Generalised Total Least Squares Problem and Solution

We mentioned earlier that, in the case where a new view is being synthesised and the positions of the control points are located by interpolating the positions of the control points in the basis views, the errors on the target view control points will not be independent of the errors on the basis views. Under these circumstances, where there may be correlation between the errors, or the errors may not be identically distributed, the problem should be solved as a generalised total least squares problem [VV91, VV89].

The generalised total least squares problem may be formulated as follows. Suppose we are given a problem of the form:

$$AX \approx B \quad , \tag{4.2.11}$$

where A is an $m \times n$ design matrix those first n_1 columns are error free, X is the $n \times d$ solution matrix that we wish to estimate and B is an $m \times d$ matrix that also contains errors. The errors in A_2 and B are correlated and of unequal size.

We can partition the design matrix A as $(A_1 \quad A_2)$, where A_1 is a $m \times n_1$ error-free matrix and A_2 is a $m \times n_2$ matrix. We can also partition the matrix X as $(X_1 \quad X_2)^T$, where the matrix X_1 is $n_1 \times d$ and X_2 is $n_2 \times d$. Assume that matrices C and D are given such that the errors in the matrix $(D^{-1})^T (A_2 \quad B) C^{-1}$ are uncorrelated and identically distributed. Then the generalised total least squares problem seeks to find $\hat{A}_2$ and $\hat{B}$ such that the error norm:

$$\left\| (D^{-1})^T \left(A_2 - \hat{A}_2 \quad B - \hat{B} \right) C^{-1} \right\|^2 \tag{4.2.12}$$

is minimised. Once matrices $\hat{A}_2$ and $\hat{B}$ have been found, such that (4.2.12) is minimised the solution for $X = (X_1 \quad X_2)^T$ is a vector that satisfies the equation [VV91]:

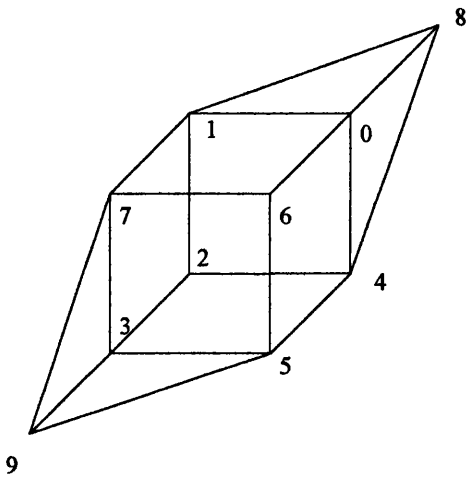
$$A_1 X_1 + \hat{A}_2 X_2 = \hat{B} \quad . \tag{4.2.13}$$

A method for solving generalised total least squares problems, based on the generalised singular value decomposition [GV96, VV89], can be found in [VV89].

4.3 Evaluation of the Least Squares and Total Least Squares Methods for Estimating the Affine Multi-view Relationships

In this section we evaluate the total least squares solution by comparing the errors on the relationships in equation (4.2.9), solved using the TLS method, with the errors on the overcomplete relationships in equation (4.2.2), which are solved as a classical least squares problem. The aim is to show that by carefully considering where the errors occur and by solving as a total least squares problem it is possible to produce a more accurate pair of multi-view relationships. The tests were carried out using synthetically generated images of a translucent geometrical test object. The vertices of the test object were used as control points. The test object consisted of ten vertices,

eight at the corners of a "unit" cube and two points at positions $(2,2,2)$ and $(-2,-2,-2)$ on the diagonal line passing through opposite corners of the cube, as illustrated in figure 4.1.



Point	Co-ordinates
0	(1, 1, 1)
1	(-1, 1, 1)
2	(-1, 1, -1)
3	(-1, -1, -1)
4	(1, 1, -1)
5	(1, -1, -1)
6	(1, -1, 1)
7	(-1, -1, 1)
8	(2, 2, 2)
9	(-2, -2, -2)

Figure 4.1. Synthetically generated geometrical test object.

Before evaluating the TLS method, we return to the Basri equations (4.2.1) and in section 4.3.1 we compare the geometric accuracy of these solutions with the overcomplete equations given in equation (4.2.2). Both the Basri equations and the overcomplete equations are solved as a least squares problem as outlined in appendix A. We then perform a series of tests with the cameras at various positions to evaluate the total least squares method described in section (4.2.1). In section 4.3.2 we place the cameras at several different distances from the test object and compare the geometric accuracy of the TLS solution and the least squares solution at each distance. At each distance the target camera is placed at the same distance from the object as the two basis cameras. The three cameras are arranged on a circular arc around the centroid of the object.

We explore, in section 4.3.3, what happens when the target camera is moved from the midpoint of the arc joining the two basis cameras. To do this we perform two tests with the target view camera moving in each of two directions. The first direction

towards one of the basis view cameras, as shown in figure 4.2, so that the target view camera still lies on the arc joining the basis view cameras, but will no longer be an equal distance from each of them. In the second test, the target view camera is moved in a direction that is initially perpendicular to the arc joining the two basis view cameras. When the target view camera is moved in this direction, the target view camera will no longer lie in the same plane as the two basis view cameras and the centroid of the object. Instead of lying on an arc, the three cameras are placed on the surface of a sphere centred on the centroid of the object.

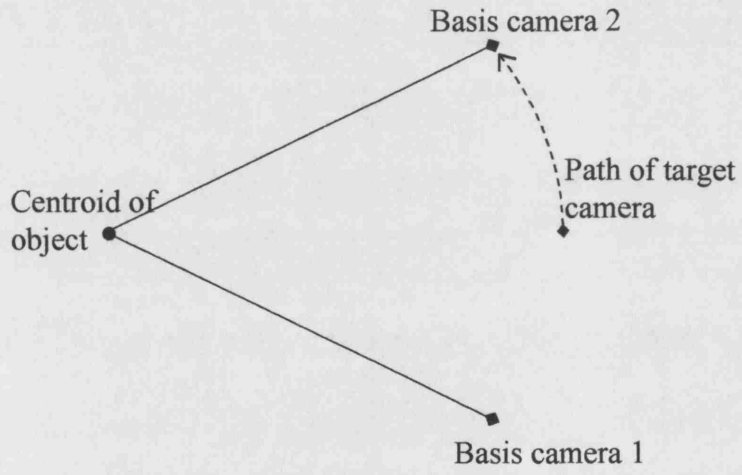


Figure 4.2. Movement of target view camera.

Since we are working with points in synthetically generated images the locations of the control points can be known exactly. This means that the errors on the relationships, in the above test, arise from the fact we are using affine imaging assumptions as an approximation to the perspective case. In practice, when using real, uncalibrated images, the positions of the control points will not be known exactly, there will be some measurement error contained within each control point. In section 4.3.4 we artificially add varying amounts of measurement errors to each of the control points in order to see how this affects the overall error on the multi-view relationships.

In the tests described above all ten control points have been used in the estimation of the multi-view relationships. In section 4.3.5 we look at how the errors

on the relationships are affected when they are estimated using different numbers of control points.

4.3.1 Comparing the Basri and Overcomplete Equations

It was mentioned in section 4.2 that, although the Basri equations (4.2.1) are exact under affine imaging conditions they can lead to large errors if the target view is close to the basis view (x'', y''), since the equations do not include a term in the co-ordinate y''. It was claimed that these large errors could be avoided by using the overcomplete set of equations (4.2.2). We will now compare these two solutions when the target view is moved between the two basis views.

For the tests the basis cameras are placed at a distance of 28 units from the centroid of the test object. The focal lengths of the cameras are chosen such that the image of the object is about 4.7 units in size, equivalent to an object occupying about 240 pixels in a 512×512 image. The basis view cameras are placed at an angle of 53° apart. Initially the target view camera is placed so that it coincides with the first basis view camera. The target view camera is then moved, by rotating around the centroid of the test object, towards the second basis view camera at roughly 5° increments. At all times the principal axis of the camera is constrained so that it passes through the centre of the test object.

For each position of the target view camera, both pairs of relationships were estimated using the least squares technique outlined in section A.1 of appendix A. Figure 4.3 shows the square root of the sum of squared errors, $\sqrt{\sum_i \varepsilon_i^2} + \sqrt{\sum_i \eta_i^2}$, on the Basri and overcomplete relationships when exact positions of the control points have been used. The errors arise from the affine imaging assumptions that have been used to approximate the perspective case. It can be seen that the errors are very similar when the target view is close to the first basis view. As the target view camera is moved away from the first basis view towards the halfway point (the point at which the target view camera is an equal distance from both basis views) the accuracy of both relationships decreases. However, the errors on the overcomplete relationships do not increase as fast as the errors on the Basri equations. When the target view camera has passed the halfway point, and is moving closer towards the second basis

view camera, the errors on the Basri equations continue to increase whereas the errors on the overcomplete equations decrease. Figure 4.4 shows the same graph but with random errors, of standard deviation 0.02 units (approximately equivalent to 1.25 pixels in a 512×512 image) added to each of the control points in each of the views. The random addition of the error was repeated two hundred times for each angle. Figure 4.4 shows the mean error and standard deviation of the errors. It can be seen from figure 4.4 that the overcomplete equations produce lower mean errors than the Basri equations.

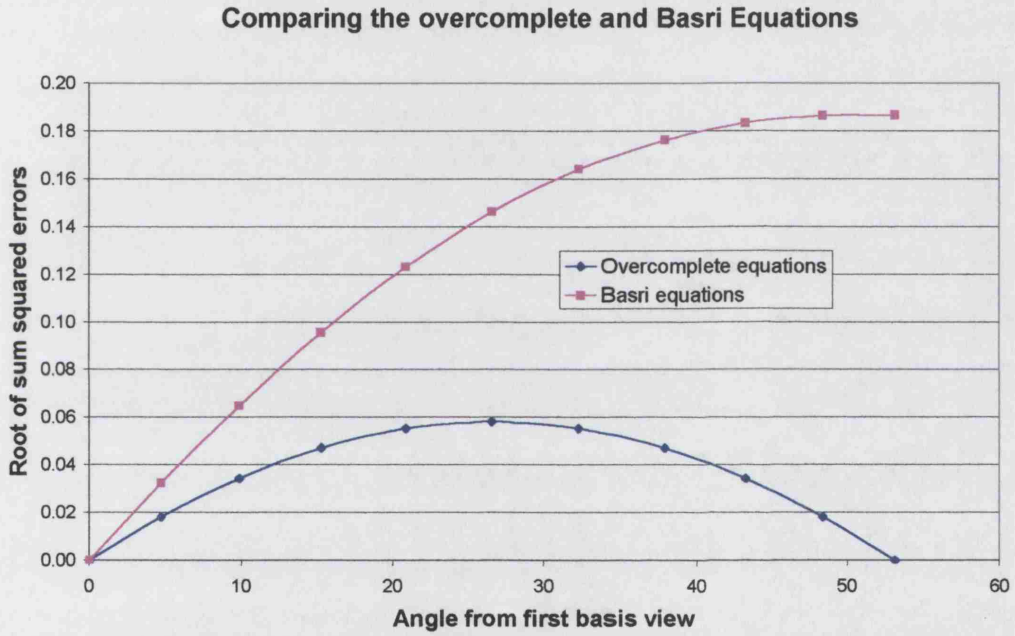


Figure 4.3. Comparing the overcomplete and Basri equations.

It can also be seen from figures 4.3 and 4.4 that when the target view is closer to the second basis view the overcomplete solution produces a pair of relationships that are much more accurate than the Basri relationships. When the target view is close to the first basis view and the exact control points are known (figure 4.3) there is little difference in the errors of the two solutions. However when the control points are known to contain measurement errors (figure 4.4) the overcomplete solution produces lower errors than the Basri relationships. When the target view is not close to the first basis view the overcomplete solution produces values that have both lower mean error and lower variation in the error. This would be expected because the overcomplete equations have an extra degree of freedom.

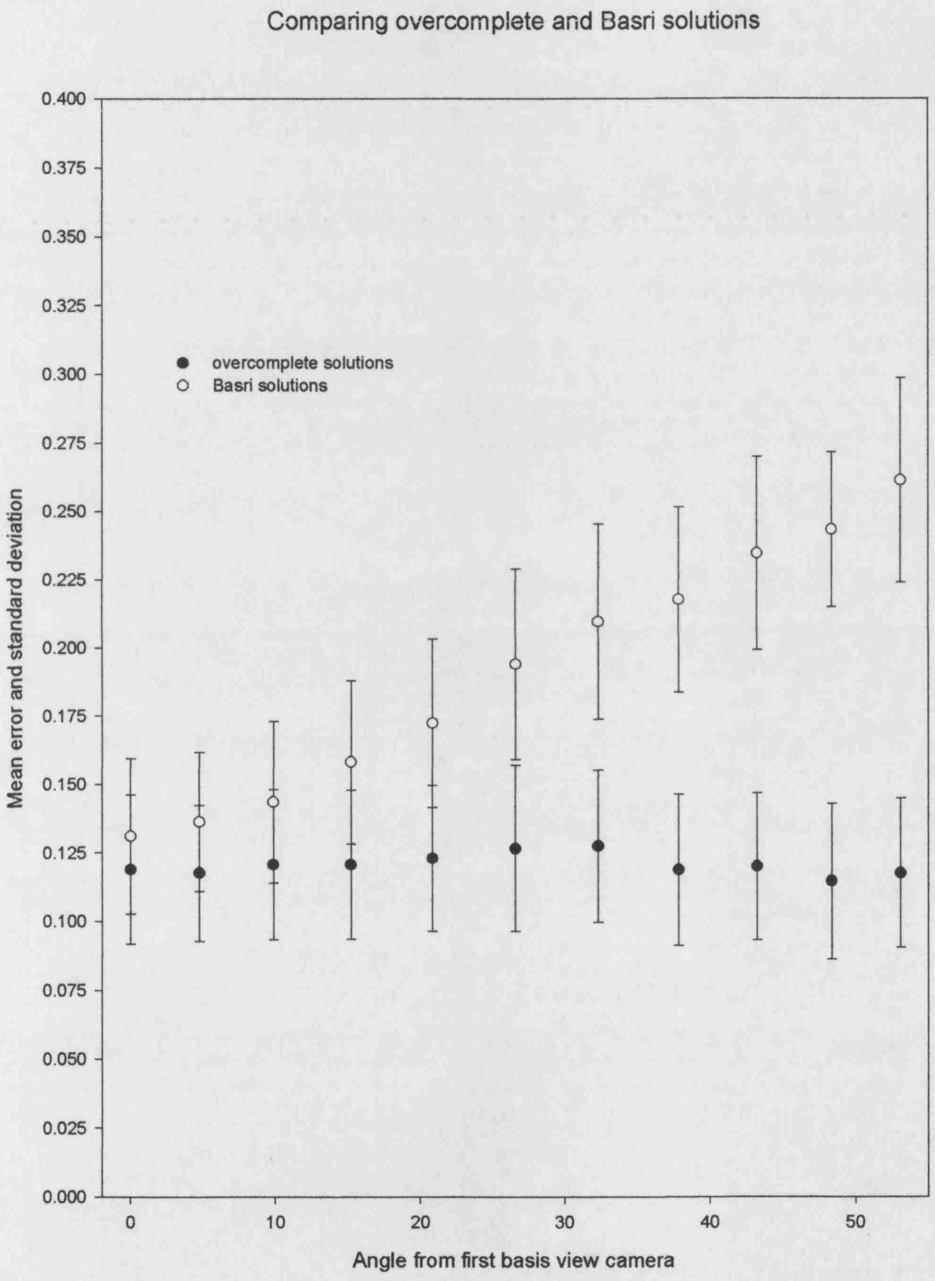


Figure 4.4. Comparing the overcomplete and Basri equations with added errors on the control points.

4.3.2 Performance of the TLS Method for Different Distances between the Cameras and the Object

We will now compare the accuracy of the overcomplete, symmetric relationships in equation (4.2.9) with the overcomplete equations (4.2.2). In this section we compare the accuracy of the two pairs of equations when the cameras are placed at different distances from the test object (in figure 4.1). A comparison of the two methods in which the camera is kept at a constant distance from the object as in section 4.3.1, above, is given in section 4.3.3. As in the previous experiments the relationships are estimated from a set of control points in synthetic images. The co-efficients in (4.2.9) are estimated using the total least squares method described in section 4.2.2 and the co-efficients in (4.2.2) by using the least squares method in appendix A. In future we will refer to the two pairs of relationships as the TLS relationships and the LS relationships respectively.

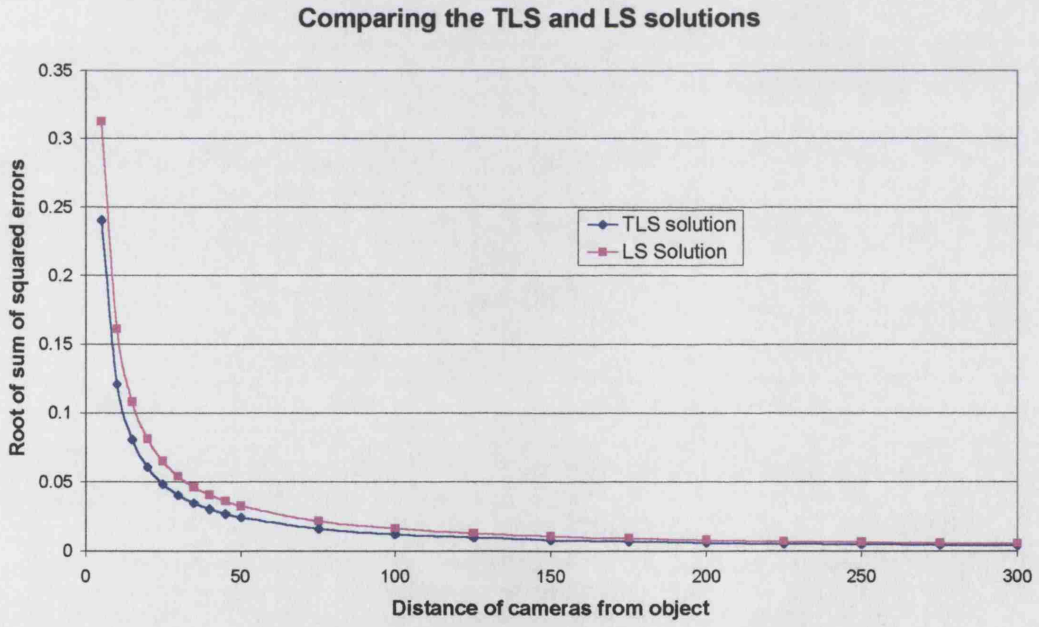


Figure 4.5. Moving the cameras towards the test object.

For this test the basis view cameras are placed at initial positions of 300 units away from the centroid of the test object, at an angle of 53° apart. The initial target view camera is also placed at a distance of 300 units from the test object and is placed midway between the two basis view cameras. The cameras were then moved on a straight line from their initial positions towards the object to a distance of 5 units from

its centre. As the cameras approach the object there will be greater perspective foreshortening and the affine assumptions will be less valid. Figure 4.5 shows the square root of the sum of squared errors on both the TLS relationships and the LS relationships. The positions of the control points used in this experiment are real and exact, we have not added any artificial errors. It can be seen from figure 4.5 that the error on the TLS relationships is always lower than for the LS relationships and the difference in the two solutions increases as the cameras are moved towards the object. Although the TLS relationships have lower errors than the LS relationships we will see in section 4.5.2 that both pairs of relationships can be used to reconstruct realistic images. It can also be seen in figure 4.5 that the errors on both pairs of relationships are increasing as the cameras are moved towards the centroid of the object. This is because of the increase in perspective effects as the cameras are moved closer to the object. The fact that the difference in the errors on the two solutions increases as the cameras move closer to the test object means that the TLS relationships are less sensitive to the perspective effects than the LS relationships.

An idea of how the perspective effects change over the range of distances chosen for the experiment is given by the values of some affine invariants [Zis92]. When the cameras are placed at large distances from the object we would expect the affine invariants to remain fairly constant since, under these conditions, the affine camera is a good approximation to the perspective camera. The affine invariants we use are of four points lying on a plane, with no three collinear, as described in section 2.4. In order to calculate the invariants the vertices of the test object are labelled 0 to 9 as show in figure 4.1. It can be seen from figure 4.1 that vertices 4, 7, 8 and 9 lie on a common plane. We choose these four points to calculate the two affine invariants, A_1 and A_2:

$$A_1 = \frac{|m_{748}|}{|m_{789}|} \quad , \qquad A_2 = \frac{|m_{749}|}{|m_{489}|} \quad , \tag{4.3.1}$$

where $m_{ijk} = (p_i, p_j, p_k)$, $p_i = (x_i, y_i, 1)^T$, and $|m|$ is the determinant of the matrix m.

Figures 4.6 and 4.7 show how the two invariants vary in the first basis view, as the camera is moved from an initial distance of 300 units from the object towards the object to within 5 units from the centroid of the object. It can be seen from these figures that there is not much variation of the invariants at large viewing distances. When the viewing distance is below 30 units there is a rapid change in the value of the

invariants owing to the large perspective effects at small viewing distances. The increase of perspective effects as the cameras are moved towards the object is consistent with the errors, on both pairs of relationships in figure 4.5, increasing as the viewing distance is decreased.

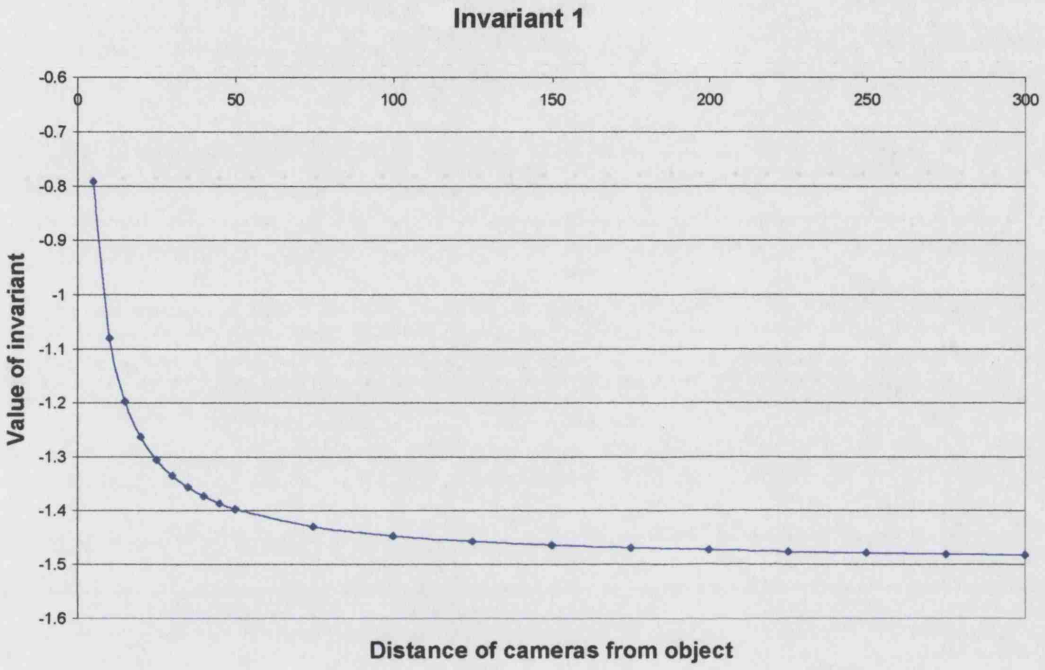


Figure 4.6. Value of affine invariant A_1.

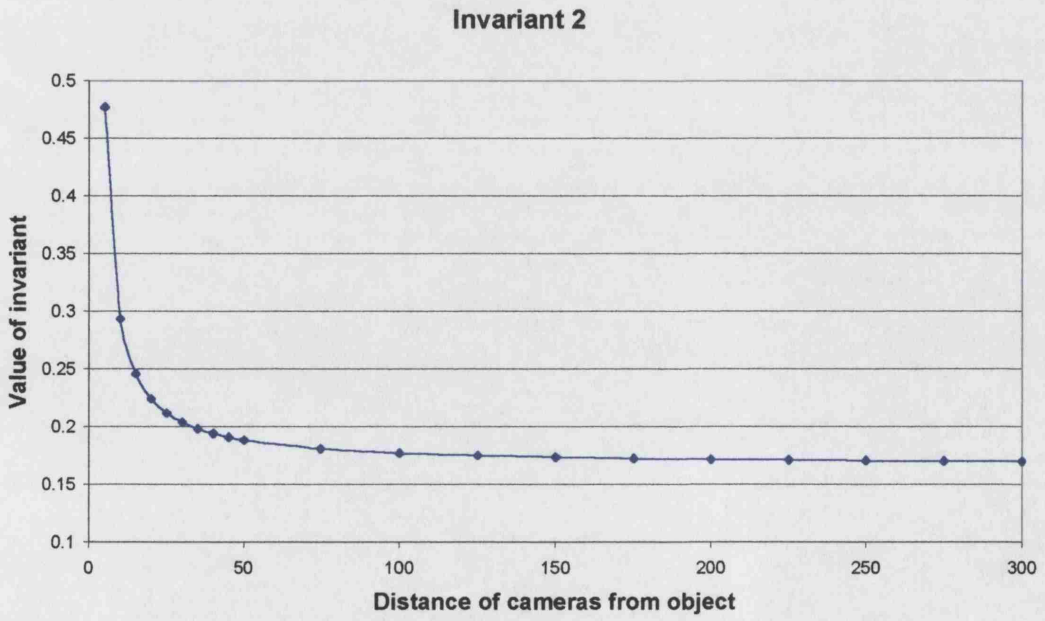


Figure 4.7. Value of affine invariant A_2.

4.3.3 Evaluation of the TLS Method as the Target View Camera is moved from the Mid-point of the Two Basis View Cameras

Throughout the distance evaluation, in the previous section, the target view camera was placed in the plane passing through the two basis view camera centres and the centroid of the object, and kept at an equal distance from both the basis view cameras. In this section we explore what happens to the errors when this constraint is removed. We perform two experiments with the target view camera moving in each of two different directions. For these experiments the basis view cameras are placed in the same positions as in the experiment in section 4.3.1. The two basis cameras are placed at an angle of about 53° apart at a distance of 28 units from the centroid of the test object. The initial target view camera is placed mid-way between the two basis view cameras, and is also at a distance of 28 units from the test object.

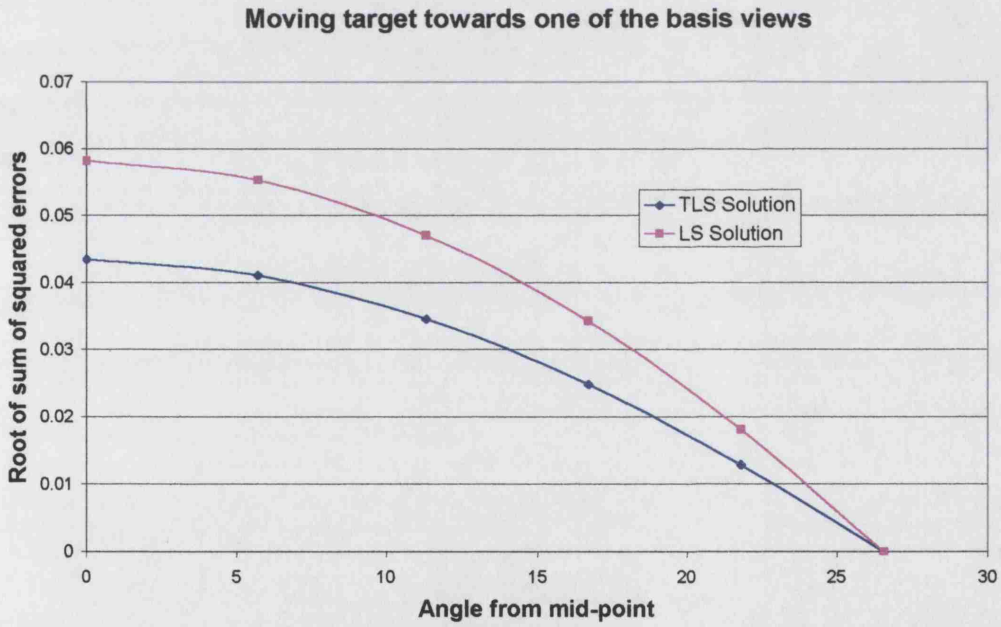


Figure 4.8. Moving the target view camera towards one of the basis views.

In the first experiment the target view camera is moved from its initial position towards the second basis view camera, as shown in figure 4.2, at intervals of roughly 5°. Figure 4.8 shows the root of the sum of squared errors of the TLS solution and the LS solution as the target view camera is moved from the mid-point towards the second basis view camera. In these experiments the exact, synthetic locations of the

control points have been used to determine the multi-view relationships therefore there are no error bars on figures 4.8 and 4.10. In sections 4.3.4 and 4.3.5 we will add random errors to the control points. It can be seen in figure 4.8 that the error on both solutions decreases as the target view camera is moved towards the basis view and that the TLS solution gives lower errors than the least squares solution. It can also be seen that the difference in error values of the two solutions decreases as the target view camera is moved closer to the basis view camera. Using the TLS solution instead of the least squares solution leads to more accurate results. This improvement is greatest when the target view camera is placed at an equal distance from the two basis view cameras.

In the second experiment the target view camera is moved from its initial position, an equal distance from each of the basis views, in the initial direction of the cameras Y-axis, as shown in figure 4.9, keeping it at an equal distance from the two basis view cameras. At each target view position the camera is rotated so that it remains fixated on the centroid of the test object. It can be seen from figure 4.10 that the TLS solution gives lower errors than the least squares solution, and that the difference in the solutions increases as the target view is moved further from the two basis views.

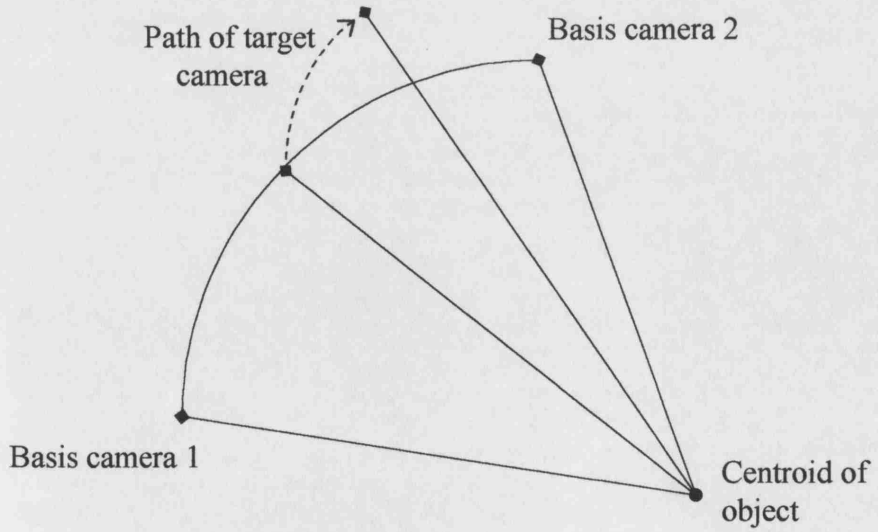


Figure 4.9. Movement of target view camera.

Figure 4.10. Moving the target view camera perpendicular to the baseline.

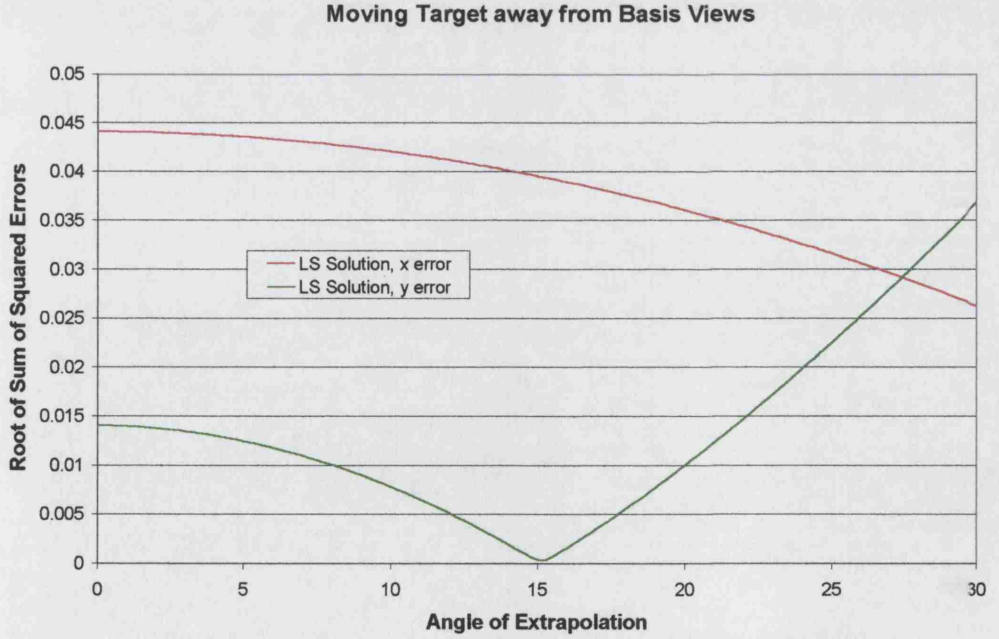


Figure 4.11. Errors on the x and y co-ordinates.

Naively we might expect the error to increase as the target view is moved away from the two basis views, whereas the graph shows that the error is initially decreasing until a distinct minimum is reached at, in this case, around 15°. In order to explain the shape of the error curves we consider the error on the LS solution and plot the error on the x and y relationships (in equation (4.2.2)) separately, as shown in figure 4.11. It can be seen from figure 4.11 that the error on both the relationships in the LS solution are initially decreasing. The errors on both the x and y relationships continue to decrease until the error on the y relationship reaches a minimum value that is close to zero. The point at which the minimum value of the y error occurs corresponds to the point where the overall error also reaches its minimum value in figure 4.10. After this point the error on the y relationship increases, and does so at a faster rate than the error on the x relationship decreases. This leads to an increase in the overall error value as can be seen in figure 4.10. In order to determine whether this initial decrease in the error values is a result of the configuration of the vertices of the test object the test was repeated for a random set of control points. Although we do not know what causes this decrease in the error values we found that it appears not to be a phenomenon completely peculiar to the choice of object. We have chosen not to investigate this phenomenon further in this thesis.

4.3.4 Adding Noise to the Control Points

In the previous experiments, in sections 4.3.2 and 4.3.3, the multi-view relationships have been determined from the exact, synthetic locations of the control points. The errors on the relationships are a result of using the affine imaging assumptions as an approximation to the perspective case. In practice the control points will inevitably contain some measurement errors.

We will now explore what happens to the errors on the relationships when we artificially add random errors to the control points. The errors were added to all control points in each view in a Gaussian distribution. Three tests were performed using added errors of standard deviations 0.005, 0.02 and 0.04 units, the equivalent of approximately 0.26, 1.25 and 2.5 pixels respectively. For each test the relationships were determined using the TLS solution and the LS solution, with the cameras positioned at several different distances from the centroid of the object. The basis

view cameras were placed at an angle of 53° apart and the target view camera was placed at an equal distance from each of the basis view cameras. At each distance the test was repeated fifty times and the mean error and the standard deviation of the error recorded.

Figures 4.12, 4.13 and 4.14 show the mean error and the standard deviation of the errors for the three different levels of noise added. The results of the LS solution have been offset slightly to the right so that the graphs can be interpreted more clearly. It can be seen from figures 4.12 to 4.14 that, as in the previous experiments, the TLS solution produces lower errors than the least squares solution. We can also see that, as the amount of noise added to the points increases, the variation in the errors also increases. The TLS solution produces lower variations in the errors than the LS solution.

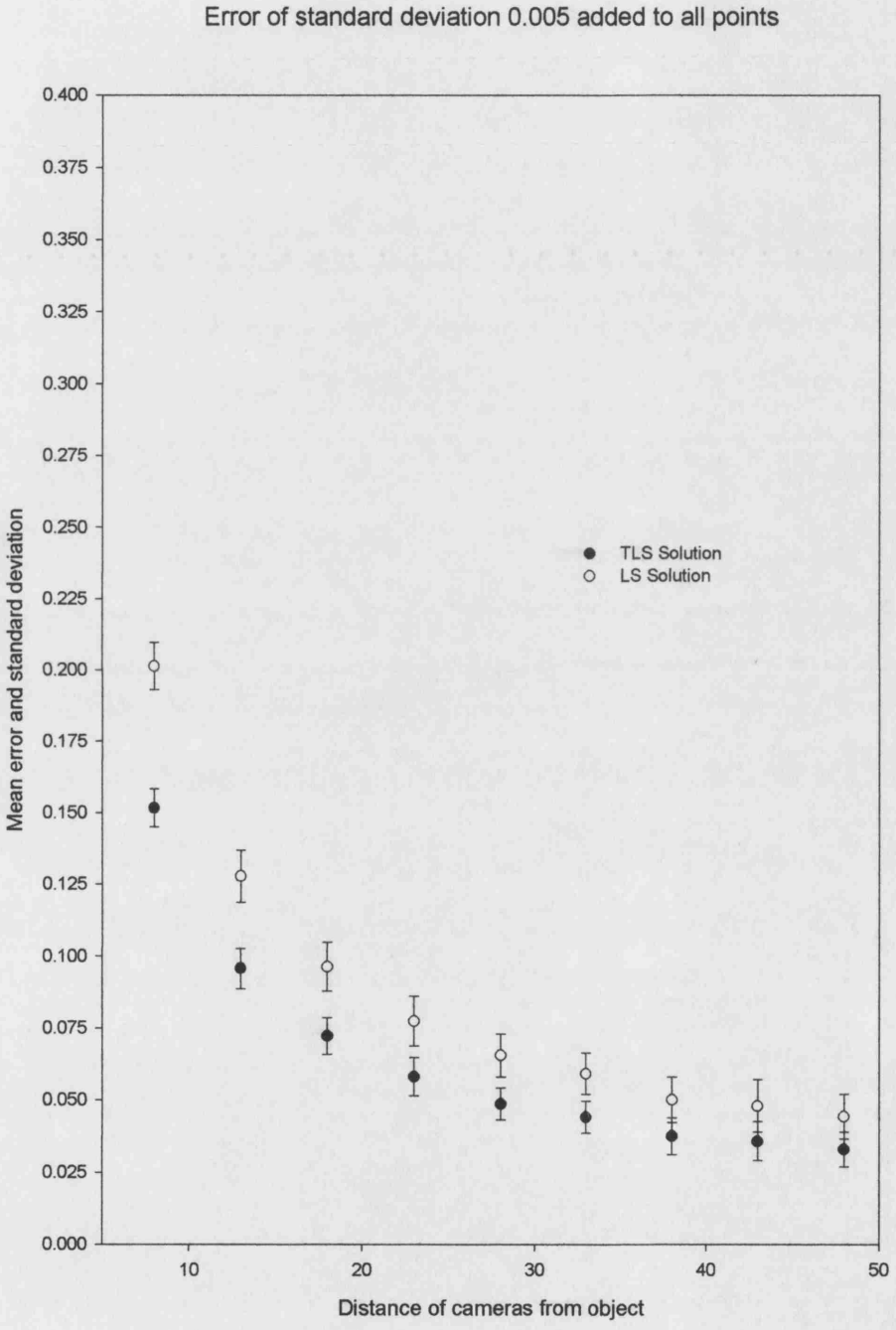


Figure 4.12. Errors of standard deviation 0.005 added to the control points.

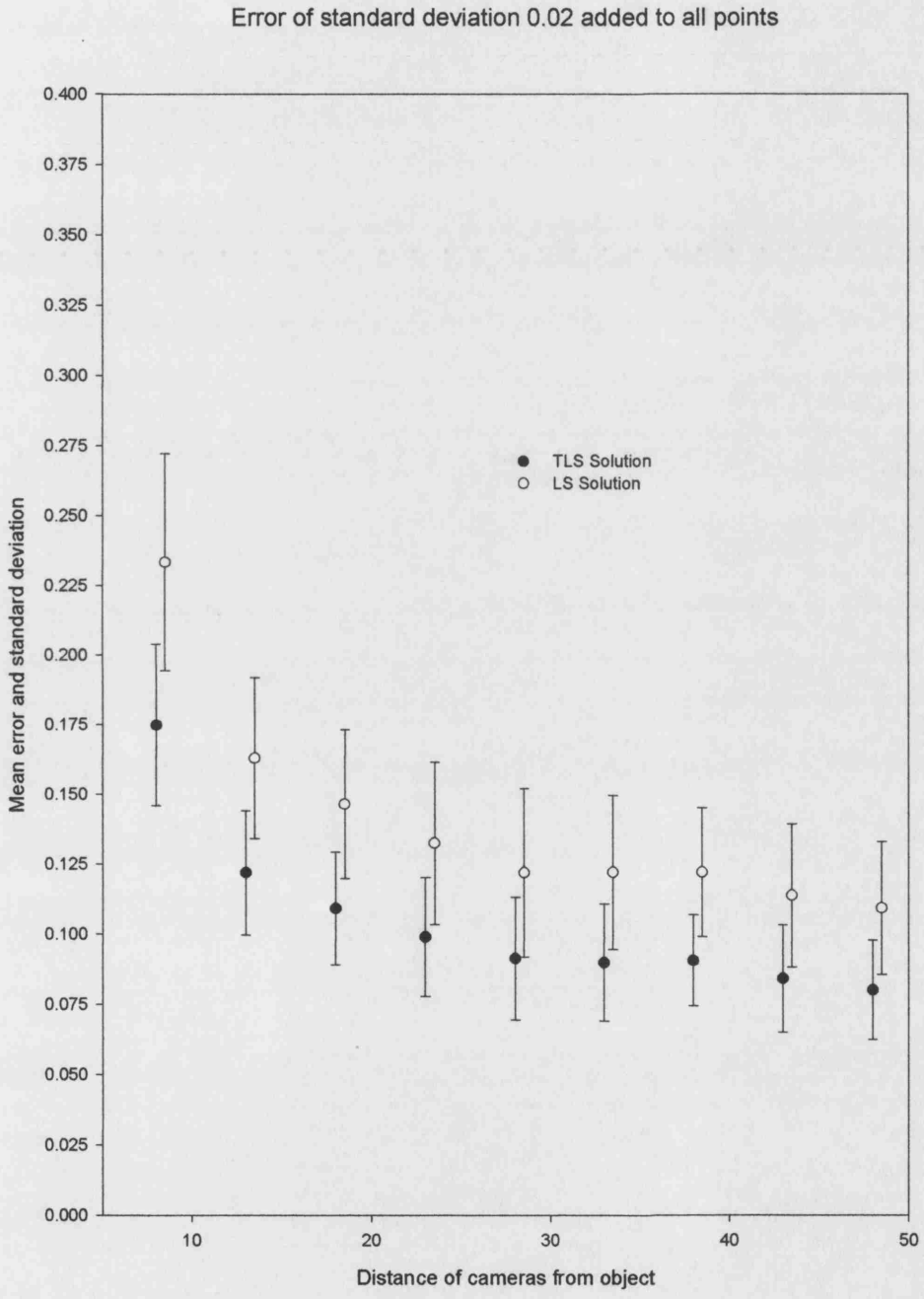


Figure 4.13. Errors of standard deviation 0.02 added to the control points.

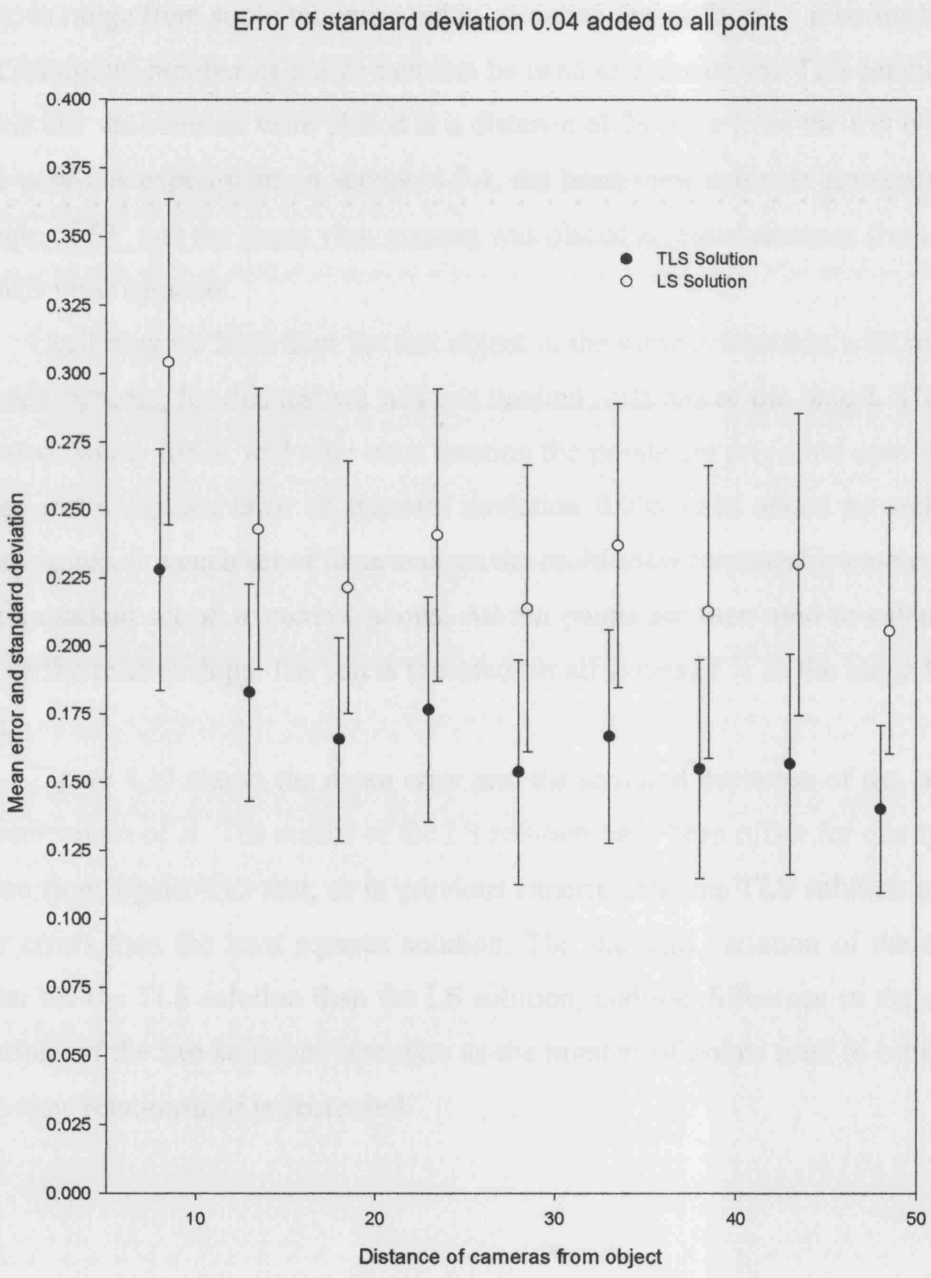


Figure 4.14. Errors of standard deviation 0.04 added to the control points.

4.3.5 Varying the Number of Points used to Estimate the Multi-view Relationships

In the previous experiments all ten control points have been used to estimate both the TLS relationships and the LS relationships. In this section varying numbers of control points, in range from six to ten, are used to calculate the multi-view relationships. Six is the minimum number of points that can be used to estimate the TLS relationships. For this test the cameras were placed at a distance of 28 units from the test object. As in the previous experiment, in section 4.3.4, the basis view cameras are separated by an angle of 53° and the target view camera was placed an equal distance from each of the basis view cameras.

Until now we have kept the test object in the same orientation with respect to the basis cameras, for this test we will use random rotations of the object. The object is rotated twenty times, and after each rotation the points are projected onto the three images and a random error of standard deviation 0.005 units added to each of the control points. For each set of three images the multi-view relationships are estimated using a random set of n control points. All ten points are then used to calculate the error on the relationships. The test is repeated for all values of n in the range from six to ten.

Figure 4.15 shows the mean error and the standard deviation of the errors for different values of n. The results of the LS solution have been offset for clarity. It can be seen from figure 4.15 that, as in previous experiments, the TLS solution produces lower errors than the least squares solution. The standard variation of the errors is smaller for the TLS solution than the LS solution, and the difference in the standard deviations of the two solutions increases as the number of points used to estimate the multi-view relationships is decreased.

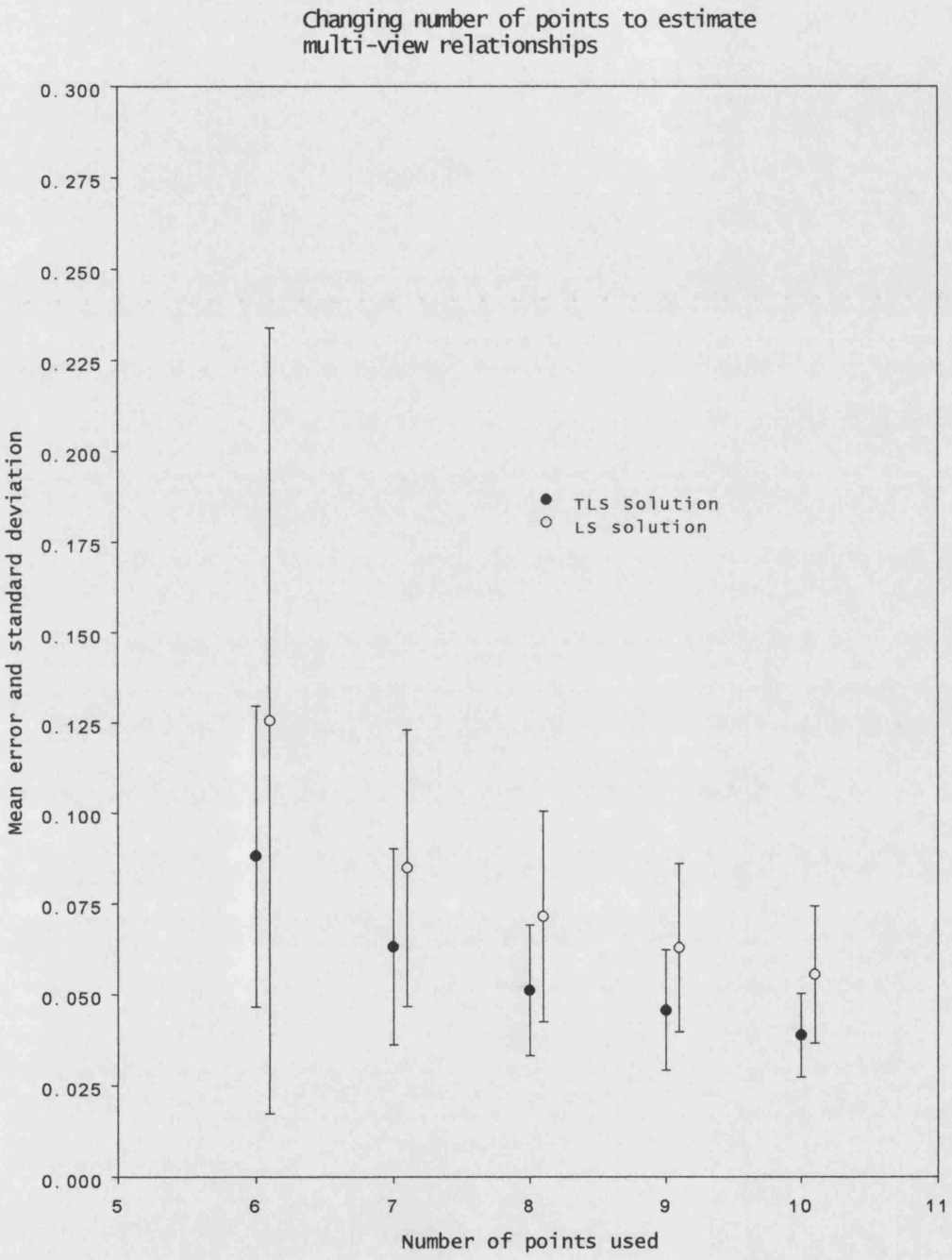


Figure 4.15. Varying the number of control points.

4.4 Encoding of Views using the Total Least Squares Solution

So far in this chapter we have looked at the methods of estimating the multi-view relationships and evaluated them by looking at the errors on the relationships. In this section we show how the total least squares relationships can be used for the encoding of views.

Section 4.4.1 discusses how the total least squares relationships can be used to transfer points from the basis views to the target view. It is important to remember the assumptions we made about where the errors occur. We assumed that all control points in all views, the target view and both basis views, are likely to contain measurement errors. The total least squares solution provides the relationships such that the sum of squared errors on the points is minimised. It may also be used to provide an estimate of what these errors are. We will show, in section 4.4.1, how we can use these estimates to correct the basis view control points. The corrected basis view control points can then be used to transfer points to the target view.

In section 4.4.2 we discuss the rendering of the target view based on the two basis views. This consists of two stages. Firstly we show how the control points can be used to obtain dense correspondence across the three images. Secondly we describe a method of setting the intensities in the target view.

4.4.1 The TLS Relationships and the Transfer of Points

In cases where the target is an existing image that we are encoding then the locations of the control points in the target will be known. Under these circumstances we may use the total least squares technique to estimate a pair of affine multi-view relationships of the form:

$$\begin{aligned} l_1\Delta x_i + l_2\Delta y_i + l_3\Delta x_i' + l_4\Delta y_i' + l_5\Delta x_i'' + l_6\Delta y_i'' &= \varepsilon_i \\ m_1\Delta x_i + m_2\Delta y_i + m_3\Delta x_i' + m_4\Delta y_i' + m_5\Delta x_i'' + m_6\Delta y_i'' &= \eta_i \end{aligned} \quad , \qquad (4.4.1)$$

where the notation Δx_i is used to represent centre of mass co-ordinates. As discussed in section 4.2.1 solutions for the co-efficient vectors $\underline{l} = (l_1, l_2, ..., l_6)^T$ and $\underline{m} = (m_1, m_2, ..., m_6)^T$, such that the sum of squared errors, $\sum_{i=1}^{n} (\varepsilon_i^2 + \eta_i^2)$, is minimised, can be found using a singular value decomposition of the design matrix D.

$$D = \begin{pmatrix} \Delta x_1 & \Delta y_1 & \Delta x_1' & \Delta y_1' & \Delta x_1'' & \Delta y_1'' \\ \Delta x_2 & \Delta y_2 & \Delta x_2' & \Delta y_2' & \Delta x_2'' & \Delta y_2'' \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \Delta x_n & \Delta y_n & \Delta x_n' & \Delta y_n' & \Delta x_n'' & \Delta y_n'' \end{pmatrix} , \qquad (4.4.2)$$

The singular value decomposition also provides an estimate of the errors on the control points. Let $D = UWV^T$ be the singular value decomposition of the design matrix as described in appendix A. If we then write the orthogonal matrices U and V as matrices of column vectors, $U = (\underline{u}_1, \underline{u}_2, ..., \underline{u}_6)$ and $V = (\underline{v}_1, \underline{v}_2, ..., \underline{v}_6)$, we may write the design matrix D as the sum of six $n \times 6$ matrices.

$$D = \sum_{j=1}^{6} \underline{u}_j w_j \underline{v}_j^T \quad . \qquad (4.4.3)$$

If we assume that the singular values are ordered in the usual way such that $w_1 \geq w_2 \geq ... \geq w_6$, then the solutions for $\underline{l}$ and $\underline{m}$ are equal to vectors $\underline{v}_6$ and $\underline{v}_5$ respectively.

We can also write the design matrix, D, which we know contains measurement errors as the sum of two matrices, $\overline{D}$ and E:

$$D = \overline{D} + E \quad , \qquad (4.4.4)$$

where we can think of $\overline{D}$ as the "true" design matrix and E as an "error" matrix. By "true" design matrix we mean the matrix $\overline{D}$ that is as close to D as possible such that the error terms ε_i and η_i in equations (4.4.1) above are equal to zero, not just minimised. In other words we wish to find the matrix $\overline{D}$ that minimises the value $\|D - \overline{D}\| = \|E\|$ and satisfies the equations:

$$\begin{aligned} \overline{D}\underline{l} &= \overline{D}\underline{v}_6 = 0 \\ \overline{D}\underline{m} &= \overline{D}\underline{v}_5 = 0 \end{aligned} \quad . \qquad (4.4.5)$$

If we substitute the expression for the design matrix D in equation (4.4.4) into equations (4.4.3) we see that:

$$\overline{D} + E = \sum_{j=1}^{6} \underline{u}_j w_j \underline{v}_j^T \quad . \qquad (4.4.6)$$

Since U and V are orthogonal matrices we know that:

$$\underline{u}_j w_j \underline{v}_j^T \underline{v}_6 = 0 \qquad \text{for all } j \neq 6 \quad , \qquad (4.4.7)$$

and similarly:

$$\underline{u}_j w_j \underline{v}_j^T \underline{v}_5 = 0 \qquad \text{for all } j \neq 5 \quad . \tag{4.4.8}$$

Since we need to satisfy equation (4.4.6), our true design matrix $\overline{D}$ cannot contain the matrices $\underline{u}_5 w_5 \underline{v}_5^T$ and $\underline{u}_6 w_6 \underline{v}_6^T$. We can separate out the matrices $\overline{D}$ and E in (4.4.6) above and write each of them as a sum of matrices of the form $\underline{u}_j w_j \underline{v}_j^T$:

$$\begin{aligned} \overline{D} &= \sum_{j=1}^{4} \underline{u}_j w_j \underline{v}_j^T \\ E &= \sum_{j=5}^{6} \underline{u}_j w_j \underline{v}_j^T \end{aligned} \tag{4.4.9}$$

The matrix $\overline{D}$ provides an estimate of the "error-free" control points such that the multi-view relationships in equation (4.4.1) are satisfied exactly. This is important when the multi-view relationships are used to encode existing views. We will call the matrix $\overline{D}$ the matrix of corrected control points.

If we are using the multi-view relationships to encode a target view as a combination of two basis views then we can store the multi-view relationships and the positions of the control points in the basis views and discard the positions of the control points in the original target view. The stored information will be enough to locate the positions of the control points in the target view by solving:

$$\begin{aligned} l_1\Delta x_i + l_2\Delta y_i + l_3\Delta x_i' + l_4\Delta y_i' + l_5\Delta x_i'' + l_6\Delta y_i'' &= 0 \\ m_1\Delta x_i + m_2\Delta y_i + m_3\Delta x_i' + m_4\Delta y_i' + m_5\Delta x_i'' + m_6\Delta y_i'' &= 0 \end{aligned} \quad . \tag{4.4.10}$$

When reconstructing a target image from the stored information the multi-view relationships are used as a pair of mapping or warping functions to transfer points from the basis views to the target view. When the relationships are used in this way it is again important to consider the assumptions that were made about the distribution of the errors. We assumed that the errors were independently and identically distributed among all control points in both the target and basis views. We have shown above that the total least squares solution also provides an estimate of the true control points, or corrected control points. If the original basis view control points, that are known to contain measurement errors, were used to transfer points to the target view then we would also be transferring the errors on the points into the target view. Instead of using the locations of the control points as given in the original basis views we should therefore instead use the corrected positions of the control points in

the basis views as provided by the TLS solution. Thus, when encoding the target view, we need to store the multi-view relationships and the locations of the corrected control points in the basis views.

4.4.2 Rendering the Target View

We have seen in previous sections that, when synthesising an encoded target view, the multi-view relationships may be used to transfer the control points from the basis views into the target view. Once the locations of the control points in the target view are known we then need a method of setting the intensities in the target image. In order to set the intensities of pixels in the target view we need to know the corresponding intensities in the two basis views. At the moment the only corresponding points we have are the control points.

We can obtain dense correspondence between the target view and each of the basis views by assuming that the object being imaged is made up of planar surfaces. To do this we use the control points in the target view to triangulate the image. Since our control points are chosen by hand we can ensure that they do lie on planar surfaces. Recent work done on locating planes in images automatically include [LAO02, LHO00 and LTAO00]. If the images are not made up of planar surfaces, for example images of a face, the triangulation should be determined such that the triangles do not include any occluding boundaries.

The triangulation can be done, for example, by using an algorithm such as Delaunay triangulation [dBVOS97]. Once we have triangulated the target image, we can then use the control points in the basis views to obtain the corresponding triangulation in each basis view. Given a set of corresponding triangles in the three images it is possible to use a piecewise linear mapping function, as described in [Gos86], inside each triangle to obtain the dense correspondence across the three images. Clearly this assumes that the triangles correspond to real planar surfaces in the 3D scene but provided the mesh formed by the triangles is sufficiently dense, this is a reasonable assumption to make. Furthermore, we could if desired use a constrained Delaunay triangulation [Peb98] to ensure that there are no prominent object material or surface boundaries inside the triangles.

However, we still do not have enough information to generate the target image. Given the correspondence between the target view and each of the basis views

we need a method of inferring the intensity to be assigned to each pixel in the target image. In general, this requires some form of interpolation or blending of the corresponding intensities in the basis views. In section 3.4.2 we gave a method of weighting the basis view intensities for the overcomplete affine relationships (4.2.2). We will now introduce a similar method for setting the target view intensities, $I(x,y)$, in terms of the intensities at the corresponding points in the two basis views, $I'(x',y')$ and $I''(x'',y'')$, when the TLS relationships are used to synthesise a target view.

To do so, we proceed in a manner analogous to that described in section 3.4.2 and define relative distance measures from the target view, $I(x,y)$, to the two basis views, $I'(x',y')$ and $I''(x'',y'')$, d' and d'' respectively, where

$$\begin{aligned} d'^2 &= l_5^2 + l_6^2 + m_5^2 + m_6^2 \\ d''^2 &= l_3^2 + l_4^2 + m_3^2 + m_4^2 \end{aligned} \qquad (4.4.11)$$

These distance measures have the required property that d' vanishes if the geometry of the target view can be obtained by an affine warping of the first basis view, $I'(x',y')$ alone. Similarly, if $I(x,y)$ can be represented as an affine transformation of $I''(x'',y'')$ then $d''=0$.

Given these distance measures we are can set the target view intensities as an interpolation of the intensities in the basis views by weighting them accordingly:

$$I = w'I' + w''I'' \quad , \qquad (4.4.12)$$

where the weights w' and w'' are given by:

$$\begin{aligned} w' &= \frac{d''^2}{d'^2 + d''^2} \\ w'' &= \frac{d'^2}{d'^2 + d''^2} \end{aligned} \qquad (4.4.13)$$

This provides a method for setting the intensities in the target view by means of an interpolation or blending of the intensities in the two basis views.

4.5 Evaluation of the Encoding of Views by using the TLS Relationships

In this section we use the total least squares relationships to synthesise target views from a pair of basis views. The multi-view relationships are estimated from a set of control points in three views. The basis views are then used to reconstruct the target view using the method described in section 4.4. For this evaluation the location of the control points and the triangulation of the images are performed manually. In section 4.5.1 the images used are of a set of coloured boxes that have been simulated using Povray. We then proceed, in section 4.5.2, to use sets of real images.

4.5.1 Simulation

A set of images of a collection of coloured boxes were simulated using Povray (http://www.povray.org/). The images were generated under orthographic projection, which is a special case of affine projection. The control points were chosen to be the vertices of the boxes, where they are visible in all three images. Where part of a box is occluded the control points were chosen to be the corners of the portion of each face of the box in the target view. The basis images are shown in figure 4.16 (a) and (b). Figure 4.16 (d) shows the reconstruction of the target image and 4.16 (c) shows the initial target view. If we compare the reconstruction (d) with the original image (c) it can be seen that the reconstruction is accurate but slightly blurred. The blurring is a result of the interpolation scheme used to render the intensities in the target view. The accuracy of the reconstruction is due to the fact that the images have been obtained using an orthographic camera, therefore the affine imaging assumptions are satisfied. If we look at the reconstruction, figure 4.16 (d), we notice that the bottom face of the red and white box is missing. This face is only visible in one of the basis views and therefore we are not able to include the vertices of this face in the set of control points. Consequently this face has been omitted from the reconstruction.

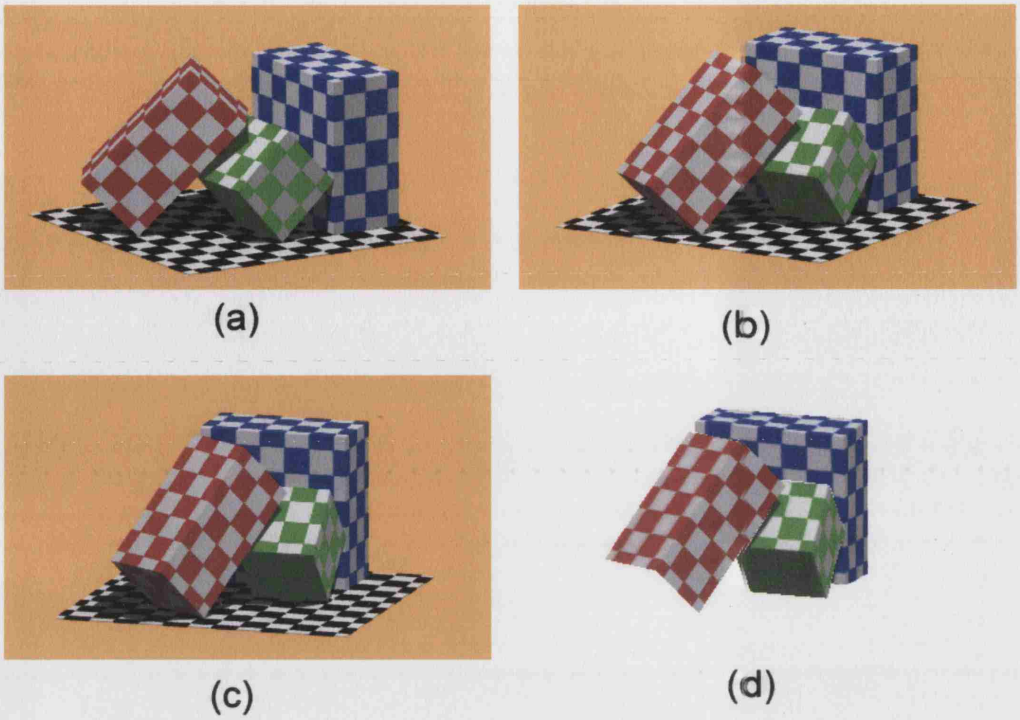


Figure 4.16. Synthetically generated boxes. Parts (a) and (b) are the basis views, (c) is the target view and (d) is the synthesised view

4.5.2 Real Images

Three sets of real images where used for the evaluation of the total least squares method of encoding. The first set of images are of a calibration object, consisting of two planar tiles fixed together at an angle of 90°. The next set of images is of a set of boxes covered in coloured wrapping paper and fixed together in a similar arrangement to the simulated boxes used in section 4.5.1. The final set are face images.

We choose to use the calibration targets as the first example because they consist of planar surfaces and do not exhibit any occlusions, both faces of the calibration targets being fully visible in all the images. Figure 4.17 (a) and (b) show the images of the calibration targets that are used as basis views. The control points are chosen to be the vertices of the two calibration tiles and are located manually. We can see that there are only six vertices, which is the minimum number of points that can be used to estimate the TLS relationships (4.2.9). When six points are used the co-efficients can be found such that the relationships are satisfied exactly, i.e. a pair of linear relationships of the form of equation (4.4.1) can be found with zero errors ε_i

and η_i on their right-hand sides. The triangulation was also performed manually and consisted of two triangles on each calibration tile.

Figure 4.17. Calibration targets. Parts (a) and (b) are the basis views, (c) is the target view and (d) is the synthesised view

The reconstruction is shown in figure 4.17 (d) whilst the original target image is shown in 4.17 (c). The reconstruction of the intensities in the target view, 4.17 (d), appears to be excellent even though the triangles that are used to interpolate the intensities have large areas. If there were greater perspective foreshortening in the images the method of reconstructing the intensities by using affine mappings might not give such good results when only a small number of triangles are used. However, given the richness of the patterns on the faces of the calibration tile, it should easily be possible to improve the results in such case by using a larger number of triangles, with smaller areas. Since there are 4 control points per plane it would also be possible to use homographies between the basis views and target view to map the target view intensities if there were greater perspective foreshortening in the images [BSG98].

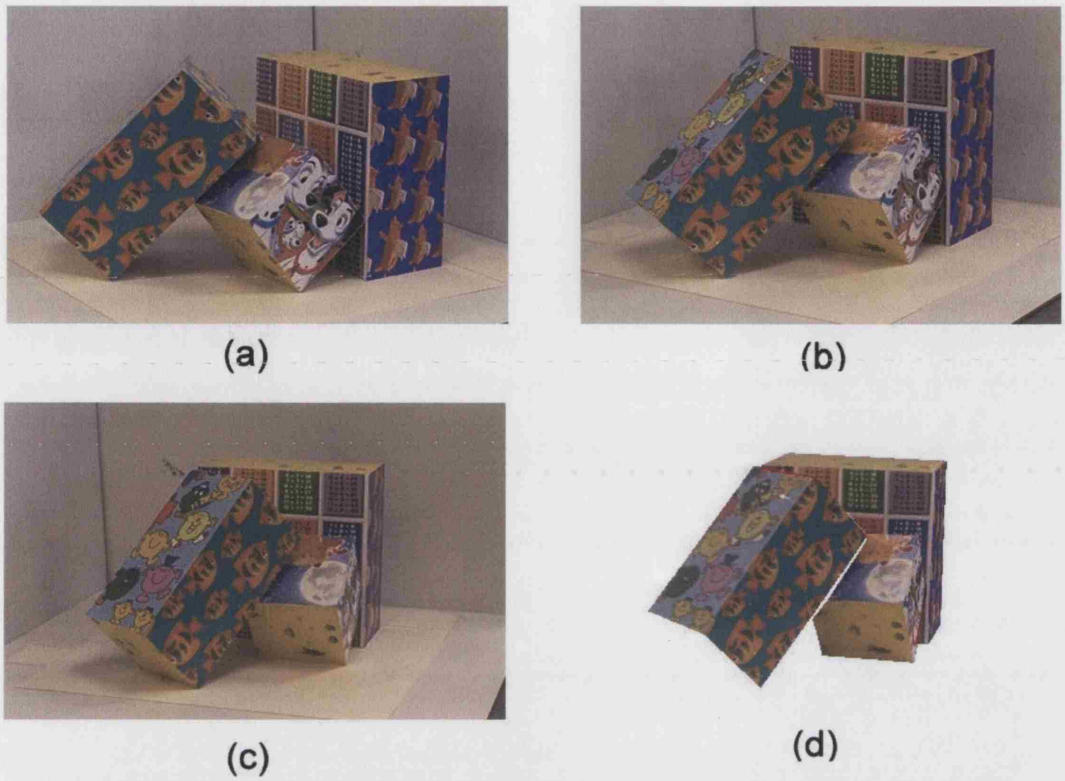


Figure 4.18. Images of an arrangement of boxes. Parts (a) and (b) are the basis views, (c) is the target view and (d) is the synthesised view.

For the box images, figure 4.18, the control points were chosen in a similar way to those for the boxes in figure 4.16. The control points are the vertices of the boxes where they are visible in all three images. Where the part of a box is occluded the control points were chosen to be the corners of the visible portion of each of the faces of the box in the target image. The triangulation was performed manually such that the interiors of the triangles did not contain any surface boundaries. The basis views are shown in figure 4.18 (a) and (b), (c) shows the original target view and (d) shows the reconstructed image.

A larger image of the reconstruction is shown in figure 4.19. The first impression of the reconstruction, figure 4.19 is that it is accurate but slightly blurred, with some noticeable artefacts around the occluding boundaries of the boxes. In particular there is a small white gap where the edge of the box nearest the cameras occludes the face of the smallest box. Along other edges we can see that some of the pixels have been coloured red. This is caused by the fact that when the approximate linear multi-view relationships are used to map points from the basis views into the target view the control points are not transferred exactly to their correct corresponding

positions. The errors in the positioning of the control points leads to a triangulation that does not completely cover the image. There are some gaps (white spaces) and overlaps (red pixels) of the edges of adjacent triangles. The case where the triangles are overlapping is easy to fix by ordering the triangles in terms of their affine depth [Ull96, Sha92]. We have coloured them red in figure 4.19 to highlight what is happening. We can remove the gaps and the overlaps by using the original hand-picked control points to reconstruct the image. The multi-view relationships would then only be needed to determine the distance measures used for weighting the basis view intensities.

Figure 4.19. Reconstruction of box image.

Finally, we use both the TLS relationships and the least squares relationships to encode and reconstruct part of a face image. The original face images are shown in figure 4.20. Parts (a) and (b) are used as the basis views and (c) is the target view. A set of 41 control points in the three images was located manually. These control points were then used to estimate the TLS relationships (4.2.9), and the least squares relationships (4.2.2). Part of the target image, figure 4.20 (c), was then reconstructed using four different methods.

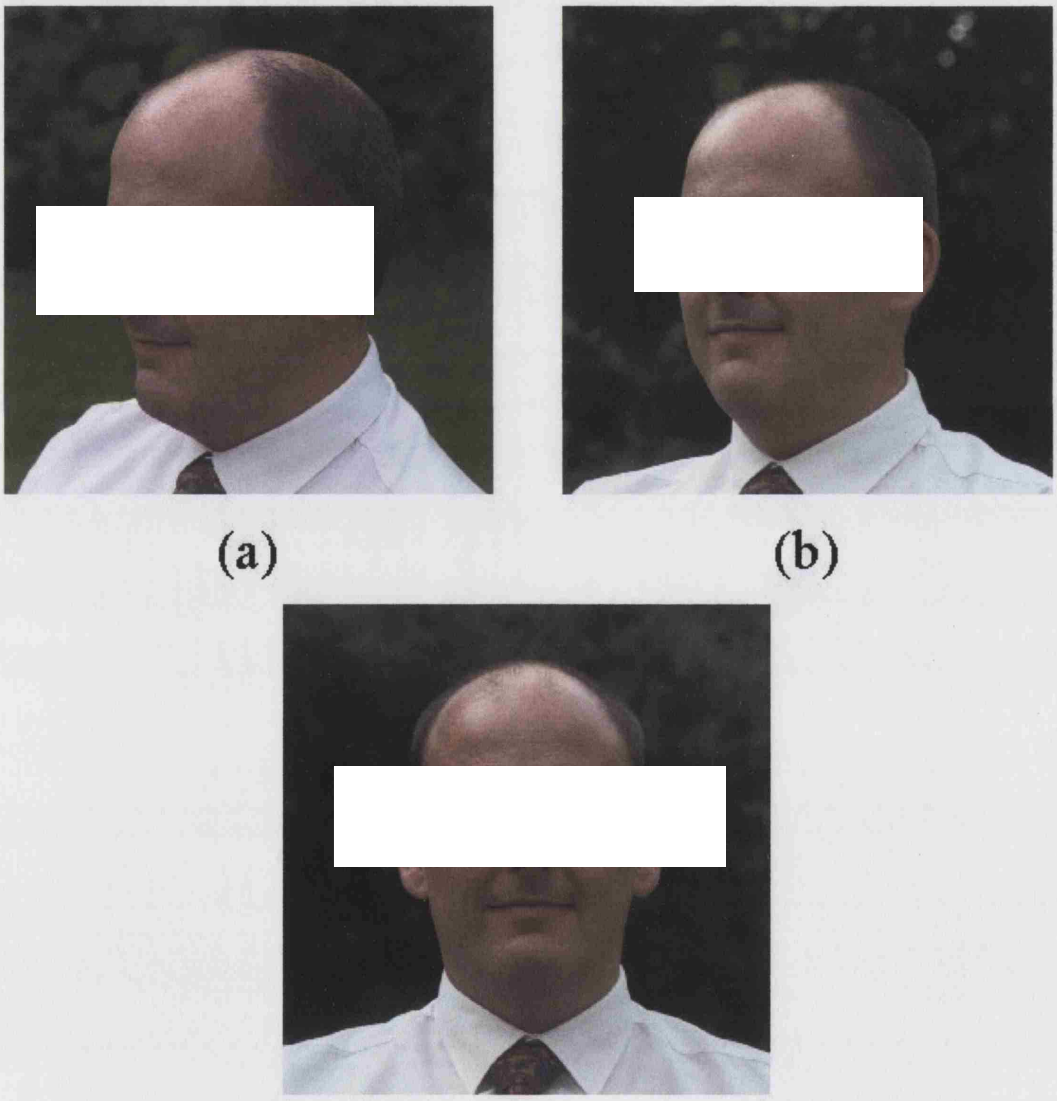
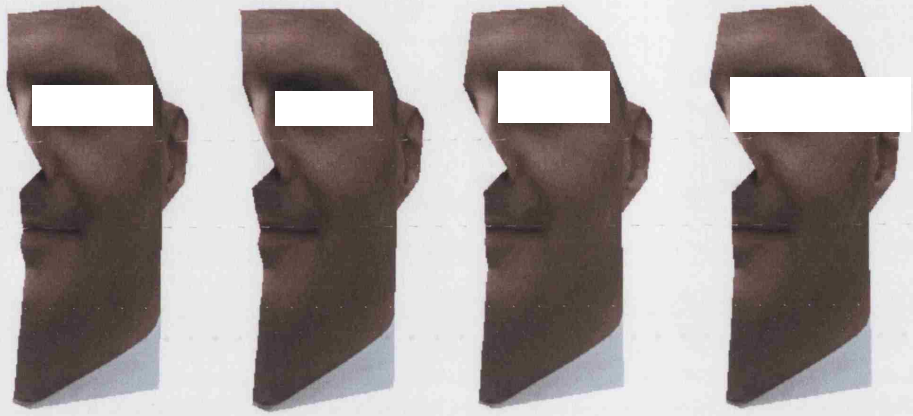


Figure 4.20. Face images.



(a) (b) (c) (d)

Figure 4.21. Reconstructed images. The points have been transferred using the least squares method in (a), the TLS method using the original basis view points in (b), the TLS method using corrected basis view points in (c). The original target view control points are used in (d) i.e., the points have not been transferred from the basis view

The four reconstructed images are shown in figure 4.21. Image (a) in figure 4.21 has been reconstructed by using the least squares relationships (4.2.2), to map control points from the basis views into the target view. The intensities have been rendered by interpolating between the two basis views using the weighting functions defined in section 3.4.2, and used previously in [BSG98, HB00a]. Figure 4.21, (b) and (c) have been reconstructed using the TLS relationships to map control points from the basis views to the target view. In image (b) we used the control points as originally located by hand in the basis views whilst in (c) we used the corrected basis view control points as described in section 4.4.1. In part (d) the control points as located in the target image itself are used for comparison. The intensities in figure 4.21 (b), (c) and (d) have all been rendered using the interpolation scheme described in section 4.4.2. The triangulation used to reconstruct the images in figure 4.21 is shown in figure 4.22.

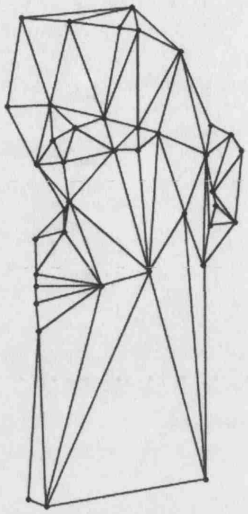


Figure 4.22. Triangulation of face image.

It can be seen from figure 4.21 that all reconstructed images are plausible and appear to be very similar. If we look closely, it can be seen that the four images have slightly different outlines owing to the control points being transferred to different positions by each of the four different methods. In order to obtain a measure of how accurate the reconstructed images are we look at two types of error measure; one measures the errors on the locations of the control points, the second measures the errors on the intensities of pixels in the target image.

4.5.3 The Error Measures

Firstly we look at the errors on the locations of the control points in the target view; that is how close the transferred control points are to those located by hand in the target image. Table 4.23 shows the sum of the absolute differences in the x and y co-ordinates between the transferred control points and the initial control points for the reconstructed images in figure 4.21 (a), (b) and (c). In figure 4.21 (d), the original hand-picked control points were used so the difference will be equal to zero.

Method of Transferring Points	Image in Figure 4.19	Sum of Absolute Errors on Control Points	Average Absolute Error per Control Point
Least Squares	(a)	182.9405	4.46196
TLS, Original Basis View Points	(b)	235.3870	5.74115
TLS, Corrected Basis View Points	(c)	37.9220	0.92493

Table 4.23. Errors in pixels on the locations of the control points.

It can be seen that when the original locations of the basis-view control points are used in the transfer process then both the least squares and TLS relationships give large error values, and the least squares solution produces a lower error value than the TLS solution. When the corrected control points, provided by the TLS solution are used in the transfer of points the total sum of squared errors on the locations of the control points in the target view is reduced dramatically and hence the sum of absolute differences likewise. This highlights the fact that it is important to consider when the errors are likely to occur, not only in the estimation of the multi-view relationships, but also in the way that they are used.

The second error that we measure is the difference between the intensity values in the reconstructed images, figure 4.21, and in the original target image, shown in figure 4.20 (c). To avoid the error measure being dominated by effects due to errors in transfer of the control points, all the reconstructed images are mapped onto the original image and the absolute difference in intensity values summed over the three colour channels are summed over all pixels in the reconstructed image. These error values are shown in table 4.24. It can be seen that all the error values are of similar magnitude and are large compared to the error values on the locations of the control points shown in table 4.23. We would expect the errors on intensities to be larger than on the locations of the control points as there are three sources of error. The first is in the location of corresponding points, the second is due to the assumption we made about the images containing planar surfaces, and the third is due to the method of weighting the intensities from the basis views. A large proportion of the error values in table 4.24 is probably because of the fact that we have (implicitly) made the assumption that the image is made up of planar surfaces when in fact we know that human faces are not made up of planar triangular facets. We will

investigate this further in the following section. We note, however, that the corrected TLS method of transferring the control points gave a lower error value than the least squares method. The highest error value is obtained when the TLS method is used without correcting the control points. As we would expect, the lowest error value is obtained when there is no warping of the control points.

Method of Transferring Points	Image in Figure 4.19	Total Sum of Absolute Differences in Intensity Values	Average Absolute difference per pixel
Least Squares	(a)	4857421	35.7766
TLS, Original Basis View Points	(b)	4860054	35.7957
TLS, Corrected Basis View Points	(c)	4853560	35.7479
No Transfer of Points	(d)	4780019	35.2062

Table 4.24. Errors in the intensity values of the reconstructed images. The average absolute difference is summed over the three colour channels (red, green and blue) each in the range 0 to 255.

4.6 Further Evaluation of the Intensity Reconstruction

We mentioned above that when we compare the intensities of the reconstructed images in figure 4.21 with the original image, figure 4.20 (c), we find that all methods of reconstruction give similar large error values. It was suggested that a large part of the error is due to the assumption that the objects in the images are made up of planar surfaces. In order to evaluate how the curvature of surfaces in the images affects the errors in the intensity values we use orthographic images of an octant of a unit sphere and similar objects made up of different numbers of planar triangular facets. The images are simulated using Povray and rendered using a random speckled texture.

The segment of the unit sphere is shown in figure 4.25 (c). Figure 4.25 (a) shows a triangulated object made up of four triangles, such that the vertices of each of the triangles all lie on the surface of a unit sphere. These vertices are used to estimate the TLS relationships. The images are then reconstructed using the actual control points and the TLS method of weighting the intensities from the basis views. For both the sphere and the triangulated object, the intensities of the reconstructed images are compared with the original images by summing the absolute difference in the three colour channels over all pixels in the image.

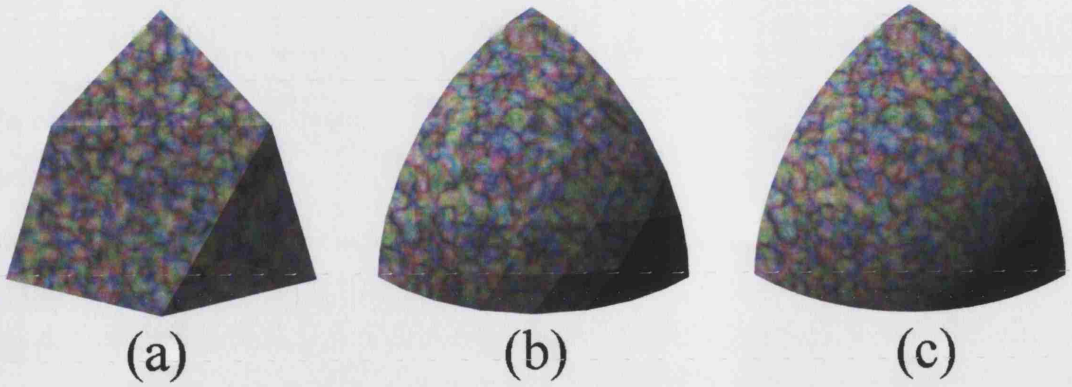


Figure 4.25. Triangulated object and a segment of a sphere.

The test was then repeated using triangulated objects consisting of 16, 64 and 256 triangles. Figure 4.25 (b) shows the triangulated object consisting of 64 triangles. Table 4.26 shows the sum of absolute difference in intensities and the average difference in intensities per pixel between the reconstructed images and the original images. It can be seen that the average error in intensity value per pixel for the triangulated object is always approximately equal to 3. We can also see that as the number of triangles used to approximate the sphere increase the error value decreases dramatically and tends to a constant value approximately equal to 3.

This test shows that if the objects in the image are not made up of planar surfaces this can lead to larger errors in the intensity values than if the images are of planar objects. We can also see from table 4.26 that, when the objects in the image are not planar, it is possible to reduce the errors in the intensity values by increasing the number triangles when triangulating the image.

Number of Triangles	Sphere		Triangulated object	
	Total sum of absolute differences	Average difference per pixel	Total sum of absolute differences	Average difference per pixel
4	4502149	56.0848	247054	3.0776
16	1970928	22.7335	257202	2.9667
36	910799	10.2997	258202	2.9199
64	522833	5.8951	260863	2.9413
100	373421	4.1962	263514	2.9612
144	313310	3.4992	266093	2.9718
196	287670	3.2171	269000	3.0083
256	274428	3.0615	267338	2.9824

Table 4.26. Errors in the intensity values of triangulated objects. The average absolute difference is summed over the three colour channels (red, green and blue) each in the range 0 to 255.

4.7 Extensions to More than Three Views

To complete this chapter we will show how it is possible to extend the TLS approach to include more than three views. Suppose that we have p views of an object or scene and we wish to find a pair of multi-view relationships that can be used to encode one of the views in terms of the others, then it is possible to seek a pair of relationships of the form:

$$\begin{aligned} l_1\Delta x_i^1 + l_2\Delta y_i^1 + l_3\Delta x_i^2 + l_4\Delta y_i^2 + ... + l_{2p-1}\Delta x_i^p + l_{2p}\Delta y_i^p &= \varepsilon_i \\ m_1\Delta x_i^1 + m_2\Delta y_i^1 + m_3\Delta x_i^2 + m_4\Delta y_i^2 + ... + m_{2p-1}\Delta x_i^p + m_{2p}\Delta y_i^p &= \eta_i \end{aligned} \quad . \qquad (4.7.1)$$

such that the sum of squared errors, $\|\underline{\varepsilon}\|^2 + \|\underline{\eta}\|^2$, is minimised. x_i^j denotes the x co-ordinate of the i^{th} control point in the j^{th} view and Δx_i^j the corresponding co-ordinate in the centre of mass reference frame. Given n corresponding control points in the p views then, provided that $n \geq 2p$, it is possible to find a solution for the co-efficient vectors $\underline{l}$ and $\underline{m}$ by using singular value decomposition of the design matrix D.

$$D = \begin{pmatrix} \Delta x_1^1 & \Delta y_1^1 & \Delta x_1^2 & \Delta y_1^2 & . & . & \Delta x_1^p & \Delta y_1^p \\ \Delta x_2^1 & \Delta y_2^1 & \Delta x_2^2 & \Delta y_2^2 & . & . & \Delta x_2^p & \Delta y_2^p \\ . & . & . & . & & & . & . \\ . & . & . & . & & & . & . \\ \Delta x_n^1 & \Delta y_n^1 & \Delta x_n^2 & \Delta y_n^2 & . & . & \Delta x_n^p & \Delta y_n^p \end{pmatrix} \qquad (4.7.2)$$

As in the case of three views, the solutions for $\underline{l}$ and $\underline{m}$ are the two singular vectors of D corresponding to the two smallest singular values.

In order to encode the intensities we define distance measures similar to those defined in the three-view case in section 4.4.2. Let us thus suppose that we wish to encode view $I^1(x^1, y^1)$, we may define distance measures from this view to each of the remaining views, k by:

$$d_k^2 = \sum_{i=3, i\neq 2k-1, 2k}^{2p} \left(l_i^2 + m_i^2\right) \quad . \qquad (4.7.3)$$

Then it is possible to encode the intensities of the target view $I^1(x^1, y^1)$ in terms of the basis views $I^2(x^2, y^2)$ to $I^p(x^p, y^p)$ as:

$$I^1(x^1, y^1) = w^2 I^2(x^2, y^2) + ... + w^p I^p(x^p, y^p) \quad , \qquad (4.7.4)$$

where the weights w^l are given by:

$$w^l = 1 - \frac{d_l^2}{d^2}, \qquad \text{where } d^2 = \frac{\sum_j d_j^2}{p-2} \quad . \tag{4.7.5}$$

This gives a method of weighting the target view intensities in terms of the basis views.

4.8 Conclusions

In this chapter we have formed a new pair of affine multi-view relationships between three views. These relationships treat each of the co-ordinates in each of the views (the target view and both basis views) in the same way; i.e., the relationships are symmetrical in each of the views. We have shown (in section 4.2.1) how it is possible to estimate these relationships using a total least squares procedure. The total least squares method allows us to treat each of the co-ordinates of in all three views in a similar fashion. By using the total least squares method we are assuming that the errors are independent and identically distributed among all of the control points. This will be the case when the control points in the basis views and the target view are located using the same method.

In section 4.3 we have evaluated the total least squares procedure by comparing the accuracy of the TLS relationships and the LS relationships. The tests were carried out using synthetically generated images of a translucent geometrical test object. We have shown that the TLS relationships produce lower error values that the LS relationships. We have also shown that the TLS relationships are less sensitive to perspective effects and to any errors on the locations of the control points.

We have described, in section 4.4, how the TLS relationships can be used to encode and reconstruct target views from a pair of basis views. In section 4.5, we have shown that it is possible to use the TLS relationships to reconstruct high quality realistic images. We have shown (in section 4.5.3) that the TLS method of reconstruction allows us to locate the target view control points more accurately than the LS method. The TLS method also gives lower errors in the rendered intensity values than the LS method.

Chapter 5

Generating Novel Views by Parameterising a Set of Sample Views

5.1 Overview

In the previous chapter we discussed how the total least squares method can be used to estimate the affine multi-view relationships, and showed how these relationships can be used to encode existing sample views as a linear combination of a pair of basis views. Until now we have assumed that either the positions of the control points or the multi-view relationships are known for the reconstructed view. In this chapter we describe a method that can be used to synthesise novel views for which the positions of the control points and the multi-view relationships are unknown. In order to do this we need to be able to predict either the positions of the control points in the novel view or the multi-view relationships between the novel view and the basis views. This means that we need a mapping from a set of parameters either to the control point positions or to the co-efficients of the multi-view relationships. The mapping needs to allow us to generate plausible novel views and because we are assuming that the cameras are uncalibrated, the mapping must be determined without any knowledge of the cameras or their locations.

We begin in section 5.2 by describing a method of finding a mapping between a set of parameters and a set of variables that will vary smoothly throughout the viewspace. In section 5.3 we explain how the method can be used to determine new TLS relationships, which can be used to synthesise novel views. In the case of the TLS relationships we cannot parameterise the co-efficients l_i and m_i directly since they do not necessarily vary smoothly throughout the viewspace. An example of when the l_i and m_i do not vary smoothly is given in section 6.1.1. Instead we choose to

parameterise the elements of the matrix R obtained from the Cholesky decomposition of the data matrix $D^T D$. This will allow us to find the both the matrices R and $D^T D$ between the basis views and the novel target view. The multi-view relationships can then be determined as the two eigenvectors of $D^T D$ that correspond to the two smallest eigenvalues. In sections 5.4 and 5.5 respectively, we show how the method can be used to parameterise the co-efficients of the LS solution and the locations of the control points. The coefficients of the LS solution (a_i and b_i) do vary smoothly throughout the viewspace so we are able to parameterise them directly. A parameterisation of the LS coefficients is given by Hansard and Buxton in [HB00a].

In section 5.6 we also evaluate the method of parameterisation when it is used to parameterise the three different sets of variables, elements of $D^T D$, the co-efficients of the LS solution (a_i and b_i) and the locations of the control points. We also discuss the advantages and disadvantages of each of the three methods.

5.2 Parameterising a Set of Sample Views

In this section we describe a method of finding a mapping between a set of variables and a set of parameters. The variables that we choose to parameterise can relate to the novel view or to the multi-view relationships between the novel view and the basis views. The only requirement we have when choosing which variables to parameterise is that they must vary smoothly throughout the viewspace. The same method of finding a mapping can be used for any of the different sets of variables. In this chapter we use three different sets of variables in the parameterisation. The three sets of variables are:

1. The elements of the matrix R obtained from the Cholesky decomposition of the data matrix $D^T D$ formed when finding the total least squares relationships.
2. The co-efficients of the least squares relationships.
3. The co-ordinates of the control points.

Here we will describe the parameterisation method for a general set of variables, E_j. We will then use the three sets of variables listed above, in sections 5.3, 5.4 and 5.5

respectively, to show how the parameterisation can be used to generate novel views. In section 5.3 the variables E_j are the elements of the upper triangular matrix obtained from the Cholesky decomposition of the data matrix $D^T D$ formed when estimating the total least squares relationships. In section 5.4 the variables E_j are the co-efficients of the least squares multi-view relationships given in equation (4.2.2). Finally in section 5.5 the variables E_j are the co-ordinates of the control points in the sample views. We will see in sections 5.5 and 5.6 that parameterising the positions of the control points has advantages over parameterising the other two sets of variables.

We will now consider the number of parameters that are needed in order to find a suitable characterisation of the sample views. A simple case, with two degrees of freedom, is where the camera is moved in a plane, whilst remaining fixated on a world point and is not rotated about its optical axis. In this case, our parameterisation will be a function of two parameters.

The general perspective camera has eleven degrees of freedom and therefore would require a function of eleven parameters. However if we assume that, *with the exception of the focal length*, the intrinsic parameters of the camera remain constant then it is possible to approximate the parameterisation functions for a perspective camera using seven parameters.

In this thesis, the multi-view relationships used in chapter 4 have been derived on the assumption that the images have been obtained under affine imaging conditions and we continue to use these assumptions in choosing a suitable parameterisation of the images. If we assume that the images are obtained using a weak-perspective camera (section 2.3.2) then translating the camera parallel to the image plane produces a shift in the image and translating the camera along its optic axis is equivalent to a change of magnification of the image. Also varying the focal length produces a scaling of the image. This means that, in the case of a weak-perspective camera, translating the camera and varying the focal length do not change the relative positions of the control points within the image and do alter the overall appearance of the image. If we assume that the images are obtained using weak-perspective cameras then the only parameters we are interested in varying are the three rotations. If we assume that the images are all taken in the same (or similar) orientation then we can assume that there is no (or very little) rotation about the optic axis of the camera. This is a reasonable assumption to make as most objects/scenes are usually only viewed in

one orientation and it will be obvious to the viewer which is the top and bottom of the image. This means that there are only two parameters that produce significant changes in the images and a suitable parameterisation of the images can be determined using two parameter functions. It is possible to find functions of any number of parameters but an increase in the number of parameters leads to an increase in the number of sample views that are needed. Therefore in the method described here we choose to write each of the variables, E_j, as a function of two parameters u and v, i.e., we are effectively parameterising a set of *surfaces*.

In the method described here we choose to fit functions for each E_j that include terms in u and v up to second order. We therefore seek a continuous function of the form:

$$E_j = a_j u^2 + b_j uv + c_j v^2 + d_j u + e_j v + f_j \quad , \qquad (5.2.1)$$

for each of the variables E_j. As we shall see, these second-order functions are complicated enough to represent the surfaces characterising the viewspace with sufficient accuracy over a significant (i.e., useful) range and require a minimum of six sample views. It is possible to include higher-order terms, although this would require a larger number of sample views.

The values of the E_j in (5.2.1) are assumed to be known for each of the sample views. If we knew the values of the parameters u and v then solving equation (5.2.1) for the co-efficients a_j to f_j would be a straightforward fitting or interpolation problem. However, since we are assuming that the cameras are uncalibrated and we do not know their locations, we do not know the values of u and v. Therefore we need a method that allows us to determine both the co-efficients a_j to f_j and the values of the parameters u and v at each of the sample views. What we have then, is a more general version of the bilinear fitting problem described in [TK92 and HZ00], which can be solved by matrix eigensolution/singular value decomposition methods.

To obtain our solution, we begin by writing out equation (5.2.1) for each of the sample views, i.

$$E_{ji} = a_j u_i^2 + b_j u_i v_i + c_j v_i^2 + d_j u_i + e_j v_i + f_j \quad , \qquad (5.2.2)$$

where $i = 1..m$, and m is the number of sample views. The E_j are the variables that we are parameterising and $j = 1..J$ where J is the number of variables that we need to parameterise. We then combine the set of equations (5.2.2) into a matrix equation.

$$\mathrm{E} = \Phi\mathrm{A} \quad , \tag{5.2.3}$$

where E is an $m \times J$ matrix of the known variables $E_j(u_i, v_i)$, written as E_{ji} for short, A is the $6 \times J$ matrix of the co-efficients $a_j ... f_j$ from (5.2.2) and Φ is an $m \times 6$ matrix depending on the parameter values u_i and v_i.

$$\mathrm{E} = \begin{pmatrix} E_{11} & E_{21} & . & . & E_{J1} \\ E_{12} & E_{22} & . & . & E_{J2} \\ E_{13} & E_{23} & . & . & E_{J3} \\ . & . & . & . & . \\ . & . & . & . & . \\ E_{1m} & E_{2m} & . & . & E_{Jm} \end{pmatrix} \quad \mathrm{A} = \begin{pmatrix} a_1 & a_2 & . & . & a_J \\ b_1 & b_2 & . & . & b_J \\ c_1 & c_2 & . & . & c_J \\ d_1 & d_2 & . & . & d_J \\ e_1 & e_2 & . & . & e_J \\ f_1 & f_2 & . & . & f_J \end{pmatrix} \tag{5.2.4}$$

$$\Phi = \begin{pmatrix} u_1^2 & u_1 v_1 & v_1^2 & u_1 & v_1 & 1 \\ u_2^2 & u_2 v_2 & v_2^2 & u_2 & v_2 & 1 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ u_m^2 & u_m v_m & v_m^2 & u_m & v_m & 1 \end{pmatrix}$$

In order to find a solution we use the requirement that we want each of the E_j to vary slowly as we change the parameter values u and v. We want each of the 2D surfaces for E_j to be as linear as possible. In other words we need to make each equation (5.2.1) as linear as possible in the parameters u and v and thus find a solution that minimises the first three rows of the matrix A in some way. Rather than simply minimising the sum of the elements in the first three rows we instead choose to minimise the principal curvature of the u,v surfaces identified by the quadratic terms. This introduces a weighting of the rows, and the value that we want to minimise, S, is the sum of the principal curvatures given by:

$$S = \sum_{j=1}^{J} \left(a_j^2 + {}^{b_j^2}\!/_{2} + c_j^2 \right) \quad . \tag{5.2.5}$$

We now wish to find a solution for (5.2.3) such that the value S is minimised. We can view this as a non-linear optimisation problem in $2m$ variables, $(u_1, v_1, u_2, ... u_m, v_m)$. However this problem is not well posed because, for any

solution, it is possible to replace u and v with an affine combination that reduces the magnitudes of the elements of the first three rows of the matrix A. For example if we scale the u and v by α it reduces the magnitudes of a_j, b_j and c_j by α^2.

We can use the following constraints on u_i and v_i to construct a well-posed problem:

$$\begin{aligned} \sum_{i=1}^{6} u_i^2 &= \sum_{i=1}^{6} v_i^2 = 1 \\ \sum_{i=1}^{6} u_i &= \sum_{i=1}^{6} v_i = 0 \\ \sum_{i=1}^{6} u_i v_i &= 0 \end{aligned} \quad . \tag{5.2.6}$$

These conditions ensure that the six views are equally distributed around the origin of the parameter space in a unique orientation.

Given a set of arbitrary parameters, $(\tilde{u}_i, \tilde{v}_i)$, it is possible to transform them into a set of parameters (u_i, v_i) that satisfy the set of constraints (5.2.6) by translating the points so that the centroid is at $(0,0)$, rotating the axes such that $\sum_{i=1}^{6} u_i v_i = 0$, and then scaling each of the axes such that $\sum_{i=1}^{6} u_i^2 = \sum_{i=1}^{6} v_i^2 = 1$.

We wish to find a solution for the matrices Φ and A in (5.2.3) such that the conditions in (5.2.6) are satisfied and that the value S in (5.2.5) is minimised.

There are $2m$ parameters (u_1, v_1) to (u_m, v_m) that we wish to determine but we have five constraints in (5.2.6), therefore we need to minimise over $2m-5$ parameters.

To begin the minimisation process we assign five of the parameters the values, $(\tilde{u}_1, \tilde{v}_1) = (0,0)$, $(\tilde{u}_2, \tilde{v}_2) = (1,0)$ and $\tilde{v}_3 = 1$. This assumes that these three views do not lie on a straight line. If all the sample views lie on a straight line we can parameterise the E_{ji} using one parameter u_i as we will show in section 6.2. We then assign arbitrary parameters values, to the remaining $2m-5$ parameters, $\tilde{u}_3$ and $(\tilde{u}_4, \tilde{v}_4)$ to $(\tilde{u}_m, \tilde{v}_m)$. The parameter values $(\tilde{u}_i, \tilde{v}_i)$ are subsequently transformed into (u_i, v_i) such that the set of conditions (5.2.6) are satisfied. The values of parameters (u_i, v_i) can

then be entered into the matrix Φ and we may use the matrices E and Φ to solve for the co-efficient matrix A. In the case where $m = 6$ the matrix A can be calculated as:

$$\mathrm{A} = \Phi^{-1}\mathrm{X} \quad . \tag{5.2.7}$$

In the case where $m > 6$ we can solve (5.2.3) as a least squares problem in the usual way as described in appendix A, with formal solution:

$$\mathrm{A} = \left(\Phi^T\Phi\right)^{-1}\Phi^T\mathrm{E} \quad . \tag{5.2.8}$$

By varying the $2m-5$ parameters, $\tilde{u}_3$ and $(\tilde{u}_4,\tilde{v}_4)$ to $(\tilde{u}_m,\tilde{v}_m)$, it is possible to find a solution for the matrices Φ and A such that the value S in (5.2.5) above is minimised. This can be done using a suitable non-linear optimisation, for example, Powell's method [PTVF93].

Once the parameter values for the m sample views and the co-efficients are known we have enough information to generate novel views. Given the parameter values, u and v, for each of the sample views we can plot their positions in the (u,v) plane. We can then choose the positions for the novel views positions in the (u,v) plane relative to the sample views. In a simple case where the camera is constrained to lie on a plane, as will be described in section 5.6.1, there are only two degrees of freedom in the parameters of the camera. In this case the parameters u and v will give the 2D positions of the cameras. In the general case there are more than two degrees of freedom for the camera matrices and the parameters u and v will not relate to any of the camera parameters. Therefore, in the general case we choose the new values for u and v by looking at the sample images and their positions in the (u,v) parameter plane and deciding which of the sample images we wish to interpolate between (or extrapolate away from). Once we have chosen new values for u and v we can substitute these into equation (5.2.1). This allows us to determine the values of the variables E_j that can be used to synthesise a novel view. As noted in section 5.1 we will now explain how this method can be used to parameterise the elements of the matrix $D^T D$.

5.3 Parameterising the Matrix $D^T D$

Recall from equation (4.2.9) that the affine multi-view relationships between three views can be expressed as a pair of relationships of the form:

$$\begin{aligned} l_1x_i + l_2y_i + l_3x_i' + l_4y_i' + l_5x_i'' + l_6y_i'' = 0 \\ m_1x_i + m_2y_i + m_3x_i' + m_4y_i' + m_5x_i'' + m_6y_i'' = 0 \end{aligned} \quad , \qquad (5.3.1)$$

where the control points in each of the three images are expressed using centre of mass co-ordinates. We have seen that when the both the target view and the basis views are known it is possible to use a set of control points to determine the multi-view relationships using the total least squares method. These multi-view relationships can then be used to encode the target view.

In this section we consider the case where we wish to synthesise a novel view by estimating the new total least squares relationships in (5.3.1). The parameterisation method allows us to form new multi-view relationships between the novel view and two of the sample views. These multi-view relationships can then be used to locate the positions of the control points in the novel view and hence synthesise the novel view.

As mentioned in section 5.1, we cannot interpolate the co-efficients the l_i and m_i, directly since they do not necessarily vary smoothly throughout the viewspace. In particular the l_i and m_i may not vary smoothly as the target view is moved towards one of the basis views until it is co-incident with that basis view. Since the multi-view relationships are obtained as eigenvectors of a matrix it is also possible that the order of the eigenvectors may alter as the views are varied throughout the viewspace. As will be shown in chapter 6, this is particularly significant where there is a degree of symmetry both in the scene and in the viewing positions.

Thus, instead of predicting the co-efficients l_i and m_i directly, we parameterise the data matrix, D^TD, that can be used to determine the multi-view relationships. We have seen earlier, in section 4.2.1, that the multi-view relationships are equal to the two singular vectors that correspond to the two smallest singular values of the $n \times 6$ design matrix D,

$$D = \begin{pmatrix} x_1 & y_1 & x_1' & y_1' & x_1'' & y_1'' \\ x_2 & y_2 & x_2' & y_2' & x_2'' & y_2'' \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_n & y_n & x_n' & y_n' & x_n'' & y_n'' \end{pmatrix} . \qquad (5.3.2)$$

As described in section 4.2.1, D is the design matrix containing the positions of the control points in both the target and the basis views, and each of the control points are expressed using centre of mass co-ordinates. Parameterising the elements of the

design matrix D is equivalent to parameterising the positions of the control points in each of the views. This will be discussed in section 5.5. In this section, instead of parameterising the design matrix D, we choose to parameterise the smaller data matrix $D^T D$.

The multi-view relationships are given by the two singular vectors of D that correspond to the two smallest singular vectors. These singular vectors of the matrix D are equal to the two eigenvectors of the matrix $D^T D$ that correspond to the two smallest eigenvectors [VV91]. The matrix $D^T D$ is a 6×6 symmetric matrix, which may be written as:

$$D^T D = \begin{pmatrix} \sum_i x_i^2 & \sum_i x_i y_i & \sum_i x_i x_i' & \sum_i x_i y_i' & \sum_i x_i x_i'' & \sum_i x_i y_i'' \\ \sum_i x_i y_i & \sum_i y_i^2 & \sum_i y_i x_i' & \sum_i y_i y_i' & \sum_i y_i x_i'' & \sum_i y_i y_i'' \\ \sum_i x_i x_i' & \sum_i y_i x_i' & \sum_i x_i'^2 & \sum_i x_i' y_i' & \sum_i x_i' x_i'' & \sum_i x_i' y_i'' \\ \sum_i x_i y_i' & \sum_i y_i y_i' & \sum_i x_i' y_i' & \sum_i y_i'^2 & \sum_i y_i' x_i'' & \sum_i y_i' y_i'' \\ \sum_i x_i x_i'' & \sum_i y_i x_i'' & \sum_i x_i' x_i'' & \sum_i y_i' x_i'' & \sum_i x_i''^2 & \sum_i x_i'' y_i'' \\ \sum_i x_i y_i'' & \sum_i y_i y_i'' & \sum_i x_i' y_i'' & \sum_i y_i' y_i'' & \sum_i x_i'' y_i'' & \sum_i y_i''^2 \end{pmatrix}, \qquad (5.3.3)$$

whereas the design matrix D is $n\times 6$, where n is the number of control points. We notice that the bottom right 4×4 sub matrix of $D^T D$ in (5.3.3) contains only information about the control points in the two basis views I' and I''. Therefore, if we assume that we are keeping the basis views constant and that they are (of course) known, then we only need to determine 11 distinct entries in the first two rows and columns of the matrix $D^T D$ for each novel view that we are synthesising. Since the elements of $D^T D$ are formed from the control points its entries will vary smoothly as the target view is moved throughout the viewspace.

We know that the matrix $D^T D$ is a positive semi-definite matrix, i.e. the eigenvalues are all greater than or equal to zero [Lüt96]. If the elements of $D^T D$ are interpolated without care, there is a danger that the matrix will loose its positive semi-definite property which could lead to a poor reconstruction of the novel view. In order to ensure that the interpolated matrix is positive semi-definite we re-arrange the matrix $D^T D$ and use a Cholesky decomposition [PTVF93] to represent $D^T D$ as $R^T R$, where R is an upper triangular matrix.

If we permute the columns of the matrix D to obtain the matrix D', where

$$D' = \begin{pmatrix} x_1' & y_1' & x_1'' & y_1'' & x_1 & y_1 \\ x_2' & y_2' & x_2'' & y_2'' & x_2 & y_2 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_n' & y_n' & x_n'' & y_n'' & x_n & y_n \end{pmatrix} , \qquad (5.3.4)$$

then the matrix $D'^T D'$ becomes:

$$D'^T D' = \begin{pmatrix} \sum_i x_i'^2 & \sum_i x_i' y_i' & \sum_i x_i' x_i'' & \sum_i x_i' y_i'' & \sum_i x_i' x_i & \sum_i x_i' y_i \\ \sum_i x_i' y_i' & \sum_i y_i'^2 & \sum_i y_i' x_i'' & \sum_i y_i' y_i'' & \sum_i y_i' x_i & \sum_i y_i' y_i \\ \sum_i x_i' x_i'' & \sum_i y_i' x_i'' & \sum_i x_i''^2 & \sum_i x_i'' y_i'' & \sum_i x_i'' x_i & \sum_i x_i'' y_i \\ \sum_i x_i' y_i'' & \sum_i y_i' y_i'' & \sum_i x_i'' y_i'' & \sum_i y_i''^2 & \sum_i y_i'' x_i & \sum_i y_i'' y_i \\ \sum_i x_i' x_i & \sum_i y_i' x_i & \sum_i x_i'' x_i & \sum_i y_i'' x_i & \sum_i x_i^2 & \sum_i x_i y_i \\ \sum_i x_i' y_i & \sum_i y_i' y_i & \sum_i x_i'' y_i & \sum_i y_i'' y_i & \sum_i x_i y_i & \sum_i y_i^2 \end{pmatrix} . \qquad (5.3.5)$$

Now the top right 4×4 sub matrix contains only data from the two basis views and therefore will remain constant as the target view is altered. If we let R be the upper triangular matrix obtained from the Cholesky decompostion (appendix C) of the matrix $D'^T D'$ such that

$$D'^T D' = R^T R \qquad , \qquad (5.3.6)$$

where

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} & R_1 & R_6 \\ 0 & r_{22} & r_{23} & r_{24} & R_2 & R_7 \\ 0 & 0 & r_{33} & r_{34} & R_3 & R_8 \\ 0 & 0 & 0 & r_{44} & R_4 & R_9 \\ 0 & 0 & 0 & 0 & R_5 & R_{10} \\ 0 & 0 & 0 & 0 & 0 & R_{11} \end{pmatrix} , \qquad (5.3.7)$$

then, as the target view is varied only entries in the last two columns of R (R_1 to R_{11}) are altered. Therefore, for each new target view, we need to determine the matrix elements R_1 to R_{11}. The Cholesky decomposition of the data matrix $D'^T D'$ allows us to ensure that the resulting $D'^T D'$ matrix will be positive semi-definite and does not alter the number of elements that we need to parameterise.

We can parameterise the elements of the matrix R using the method described in section 5.2. The elements that we are parmeterising are the eleven elements R_1 to

R_{11} in the matrix (5.3.7). The E_{ji} in (5.2.2) thus become the elements $R_{j,i}$, where $j = 1..11$ and $i = 1..m$ where m is the number of sample target views. The matrices E and A in (5.2.3) and (5.2.4) correspondingly become:

$$\mathrm{E} = \begin{pmatrix} R_{1,1} & R_{2,1} & . & . & R_{11,1} \\ R_{1,2} & R_{2,2} & . & . & R_{11,2} \\ R_{1,3} & R_{2,3} & . & . & R_{11,3} \\ . & . & . & . & . \\ . & . & . & . & . \\ R_{1,m} & R_{2,m} & . & . & R_{11,m} \end{pmatrix} \text{ and } \mathrm{A} = \begin{pmatrix} a_1 & a_2 & . & . & a_{11} \\ b_1 & b_2 & . & . & b_{11} \\ c_1 & c_2 & . & . & c_{11} \\ d_1 & d_2 & . & . & d_{11} \\ e_1 & e_2 & . & . & e_{11} \\ f_1 & f_2 & . & . & f_{11} \end{pmatrix} . \qquad (5.3.8)$$

The parameterisation provides us with an equation of the form:

$$R_j = a_j u^2 + b_j uv + c_j v^2 + d_j u + e_j v + f_j \quad , \qquad (5.3.9)$$

for each of the eleven elements R_j with $R_{j,i}$ evaluated at each of the m sample target views corresponding to view parameters (u_i, v_i). We can then use this information to generate novel views. We first choose the position of the novel view in the 2D parameter space (u,v) relative to the m target views. Once we have chosen the new values for u and v we can then calculate the eleven elements, R_1 to R_{11}, of the upper triangular matrix R by using equation (5.3.9). These values can then be inserted into the matrix R in (5.3.7) from which we can calculate the data matrix $D'^T D'$ according to (5.3.6).

The pair of multi-view relationships between the basis views and the novel target view are then the eigenvectors of $D'^T D'$ that correspond to the two smallest eigenvalues. It should be noted that, since we initially permuted the columns of the design matrix D to obtain D' and hence $D'^T D'$, the eigenvectors provide two multi-view relationships of the form:

$$\begin{aligned} l_1 x_i' + l_2 y_i' + l_3 x_i'' + l_4 y_i'' + l_5 x_i + l_6 y_i = 0 \\ m_1 x_i' + m_2 y_i' + m_3 x_i'' + m_4 y_i'' + m_5 x_i + m_6 y_i = 0 \end{aligned} \quad , \qquad (5.3.10)$$

where the co-efficient of the target view co-ordinates are now the last two entries in the two eigenvectors.

Once the multi-view relationships have been determined, they can be used to locate the positions of the control points in the novel view and the intensities can be rendered using the method described in section 4.4.2.

It should be noted that the Cholesky decomposition can only be performed on matrices that are positive definite. In the case where the target view is coincident with one of the basis views, the resulting $D'^T D'$ matrix will be positive semi-definite. The $D'^T D'$ matrix will also be positive semi-definite if only six control points are used. It is therefore important that the sample views that are chosen in order to parameterise $D'^T D'$ are distinct from the basis views and that we use a minimum of seven control points. This means that we actually require a minimum of eight initial sample views, two to use as basis views and a minimum of six for the parameterisation.

In section 4.4.1 we discussed the use of the TLS relationships to transfer points from a pair of basis views into a target view. It was mentioned that the TLS solution provides an estimate of the errors on each of the control points in both the basis views and the target view and that we should consider all these errors when transferring points. If we are generating a novel view by the method just described above, the positions of the control points in the novel target view are found by using the multi-view relationships to transfer points from the two basis without correcting for errors in the basis view control points. In fact, it is not possible to apply such corrections when, as here, we do not know the locations of the control points in the novel target view. Thus, if we wish to generate a novel view by interpolating the multi-view relationships, it may be better to use the LS relationships rather than the TLS relationships (see section 4.4).

5.4 Parameterisation of the Co-efficients of the LS Solution

In section 5.3 we described how it is possible to use the method of section 5.2 to parameterise a set of views in terms of the elements of the data matrix, $D^T D$. The same method can also be used to parameterise the co-efficients of the LS relationships. If we express our control points using centre of mass co-ordinates as described in section 3.3.1, we can write our pair of LS relationships as:

$$\begin{aligned} x &= \alpha_1 x' + \alpha_2 y' + \alpha_3 x'' + \alpha_4 y'' \\ y &= \beta_1 x' + \beta_2 y' + \beta_3 x'' + \beta_4 y'' \end{aligned} \qquad (5.4.1)$$

In the case of the TLS relationships it is necessary to interpolate the elements of the data matrix rather than the actual co-efficients, l_k and m_k in equation (5.3.1), since the latter do not necessarily vary smoothly throughout the viewspace. When

using the LS relationships the co-efficient vectors $\underline{\alpha}$ and $\underline{\beta}$ can be written (as shown in appendix A) as:

$$\begin{aligned} \underline{\alpha} &= \left(D^T D\right)^{-1} D^T \underline{x} \\ \underline{\beta} &= \left(D^T D\right)^{-1} D^T \underline{y} \end{aligned} \quad , \tag{5.4.2}$$

where, in this case the design matrix D only contains information from the basis views and $\underline{x}$ and $\underline{y}$ are the vectors of the control points in the sample target views $i = 1...m$. The same basis views are used for each target view. The design matrix D therefore remains constant as the target view is varied and the positions of the control points will vary smoothly as the camera is moved around in the viewspace. The co-efficient vectors $\underline{\alpha}$ and $\underline{\beta}$ will therefore also vary smoothly. Thus, when using the LS relationships we are able to parameterise the co-efficients, α_k and β_k directly. There are now only eight elements to parameterise as opposed to eleven in the parameterisation of the TLS relationships described in section 5.3.

The eight LS co-efficients may be parameterised using the method described in section 5.2. In this case, the elements E_{ji} in (5.2.2) become the coefficients α_k and β_k for each of the sample target views. The matrices E and A in (5.2.3) and (5.2.4) are now given by:

$$\mathrm{E} = \begin{pmatrix} \alpha_{11} & \alpha_{21} & \alpha_{31} & \alpha_{41} & \beta_{11} & \beta_{21} & \beta_{31} & \beta_{41} \\ \alpha_{12} & \alpha_{22} & \alpha_{32} & \alpha_{42} & \beta_{12} & \beta_{22} & \beta_{32} & \beta_{42} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ \alpha_{1m} & \alpha_{2m} & \alpha_{3m} & \alpha_{4m} & \beta_{1m} & \beta_{2m} & \beta_{3m} & \beta_{4m} \end{pmatrix} \tag{5.4.3}$$

and, $$\mathrm{A} = \begin{pmatrix} a_1 & a_2 & . & . & a_8 \\ b_1 & b_2 & . & . & b_8 \\ c_1 & c_2 & . & . & c_8 \\ d_1 & d_2 & . & . & d_8 \\ e_1 & e_2 & . & . & e_8 \\ f_1 & f_2 & . & . & f_8 \end{pmatrix} \quad . \tag{5.4.4}$$

Thus, in this case, E is the $m \times 8$ matrix containing the co-efficients α_k and β_k for each of the target views $i = 1...m$ and A is the 6×8 matrix of the co-efficients a_j to f_j.

We can then solve for the matrices Φ and A in (5.2.3) using the method described in section 5.2. By using the LS relationships instead of the TLS relationships we can reduce the number of elements we need to parameterise. We also avoid the problem of not being able to correct the basis view control points because the LS solution assumes that all the error is on the control points in the target view.

In order to parameterise the co-efficients α_k and β_k we again need a minimum of eight sample views. Six of these views will be used as the target views and the other two as the basis views.

5.5 Parameterisation of the Control Points

The method of parameterisation described in section 5.2 can also be used to interpolate each of the control points between the views. Unlike the two previous cases, we do not need to select two of the sample views to be the basis views. The choice of the basis views can be postponed as described below. Only six sample views are therefore needed to find a parameterisation, whereas we needed eight views to parameterise the data matrix $D^T D$ or the LS relationships. Once we have located the positions of the control points in the novel view we can then choose any of the sample views to render the intensities. This is particularly useful when some of the surfaces are occluded in some of the views.

Given a set of m sample views and a set of n control points in those views we can write each co-ordinate of every control point (x_j, y_j) as a function of the form given in (5.2.2).

$$\begin{aligned} x_{ji} &= a_j u_i^2 + b_j u_i v_i + c_j v_i^2 + d_j u_i + e_j v_i + f_j \\ y_{ji} &= a_{j+n} u_i^2 + b_{j+n} u_i v_i + c_{j+n} v_i^2 + d_{j+n} u_i + e_{j+n} v_i + f_{j+n} \end{aligned} \quad , \qquad (5.5.1)$$

where x_{ji} and y_{ji} denote respectively the x and y co-ordinates of the j^{th} control point in the i^{th} view.

In this case our matrices E and A in (5.2.3) are:

$$\mathrm{E} = \begin{pmatrix} x_{11} & x_{21} & . & . & x_{n1} & y_{11} & . & . & y_{n1} \\ x_{12} & x_{22} & . & . & x_{n2} & y_{12} & . & . & y_{n2} \\ . & . & & & . & . & & & . \\ . & . & & & . & . & & & . \\ x_{1m} & x_{2m} & . & . & x_{nm} & y_{1m} & . & . & y_{nm} \end{pmatrix} \qquad (5.5.2)$$

and,
$$A = \begin{pmatrix} a_1 & a_2 & . & . & a_{2n} \\ b_1 & b_2 & . & . & b_{2n} \\ c_1 & c_2 & . & . & c_{2n} \\ d_1 & d_2 & . & . & d_{2n} \\ e_1 & e_2 & . & . & e_{2n} \\ f_1 & f_2 & . & . & f_{2n} \end{pmatrix} \quad . \qquad (5.5.3)$$

Equation (5.2.3) can then be solved using the method described in section 5.2.

The disadvantage of parameterising the control points instead of the multi-view relationships is that we have more elements to parameterise. If we parameterise the control points in this way, then we are parameterising both the x and y co-ordinates of each of the control points, that is $2n$ elements are parameterised as opposed to eleven matrix elements in the case of the TLS relationships and eight co-efficients in the case of the LS relationships. However, parameterising the control points does have advantages. As mentioned above, when parameterising the control points we do not need to select the basis views in advance. In the previous two cases we have been determining the multi-view relationships between a pair of basis views and a novel view. This requires that the same two basis views will be used to render each novel view. When synthesising a novel view by parameterising each of the control points we can use any of the sample views to render the intensities in the novel view. This also means that we can find a parameterisation using a minimum of six sample views, whereas the previous two cases required eight sample views (two to be used as the basis views and six target views).

5.6 Evaluation

So far in this chapter we have described a method of parameterising a set of sample views and shown that it can be used to parameterise the views in terms of three different sets of variables. We can parameterise either the elements of the data matrix $D^T D$, formed when estimating the TLS relationships, the co-efficients of the LS relationships, or the positions of the control points. We will now use the method in section 5.2 to parameterise each of the three sets of variables as described in sections 5.3 to 5.5 and then to synthesise and compare the accuracy of novel views.

We begin, in section 5.6.1, with a set of sample images that are synthetic images obtained using Povray. In section 5.6.2 we use a set of images of a pair of

calibration targets. Finally in section 5.6.3 we use synthesise a novel view of a face image.

5.6.1 Synthetically Generated Objects

In this section we use synthetically generated images of a collection of three cubes that have been rendered using Povray. The images are 500×500 pixels and the control points were taken to be the eight vertices of each of the three cubes. Since the images are synthesised we know the exact location of each control point. All twenty-four vertices are used as control points even though not all vertices are visible in the images. The images are synthesised using a perspective camera with constant focal length. The camera is constrained to lie on a fixed plane and there is no rotation about the optical axis. The synthetic object consists of three cubes of side length 1.4 positioned such that they are centred at a distance of 1.7 units along the positive X, Y and Z axes. The camera is rotated around the fixation point, located at $(0,0,0)$.

When we parameterise the data matrix $D^T D$ (section 5.3) and the co-efficients of the LS relationships (section 5.4) we use eight sample views. Figure 5.1 shows the six sample views that are taken as the target views and the two basis views are shown in figure 5.2. As mentioned earlier the camera is constrained to lie on a fixed plane. The positions of the eight sample views in this plane are shown in figure 5.3, the six target view are represented by black dots and the two basis views are shown using green dots.

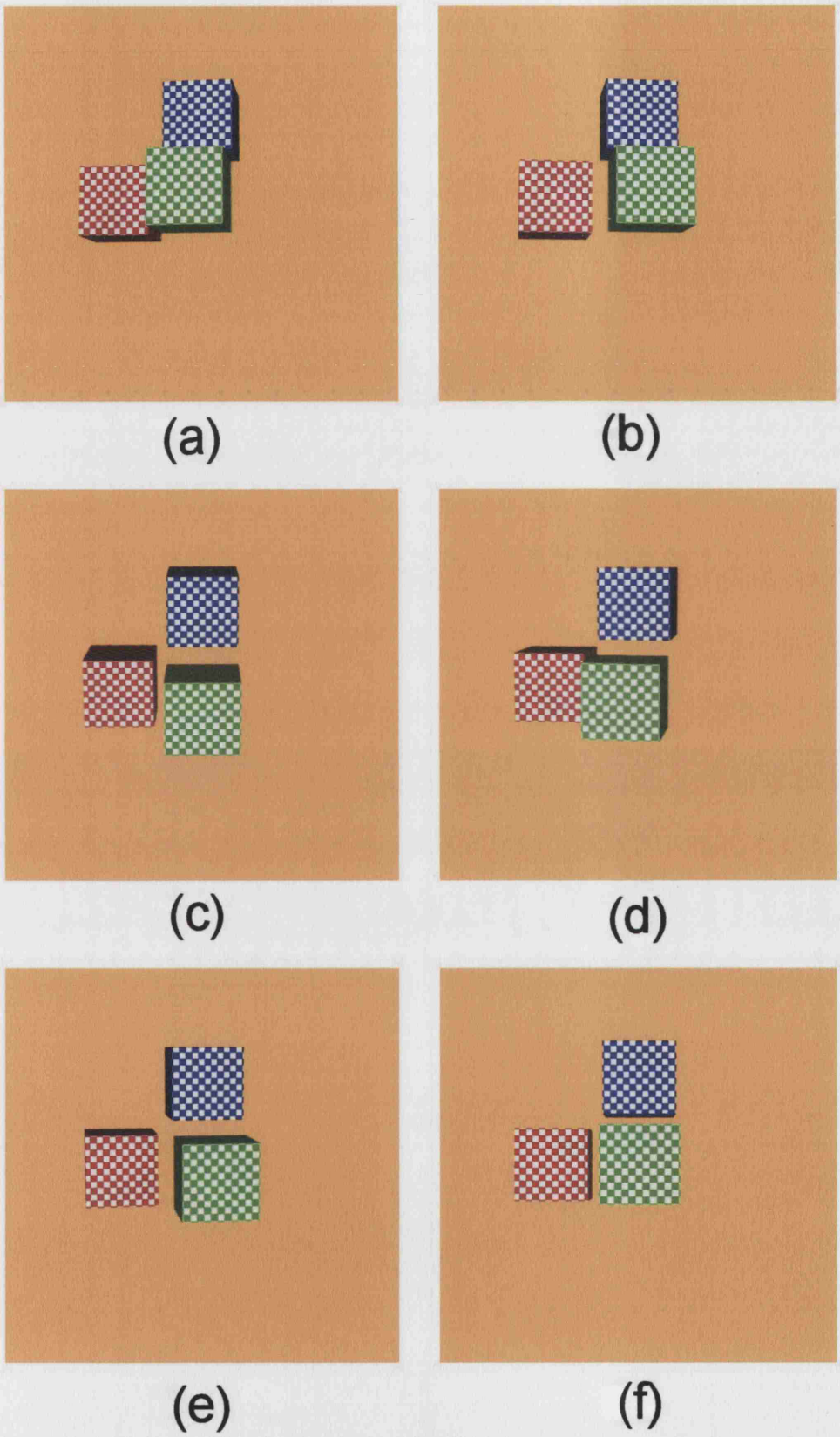


Figure 5.1. The sample target views.

As described in section 5.3, to parameterise the matrix $D^T D$ we first re-arrange the entries to obtain the matrix $D'^T D'$ and then use the Cholesky decomposition to represent the matrix as $R^T R$, where R is an upper triangular 6×6 matrix. The matrix R is calculated using each of the images in figure 5.1 as the target views and the images in 5.2 as the basis views. The target views are then parameterised using the method described in sections 5.2 and 5.3. The positions of the six target views in the 2D parameter plane obtained by this procedure are shown as black dots in figure 5.4. It can be seen that the positions of the target views relative to each other in figure 5.4 are similar to the relative positions of the actual target views in figure 5.3. The parameterisation has placed the views in a sensible arrangement in the 2D parameter plane (u, v).

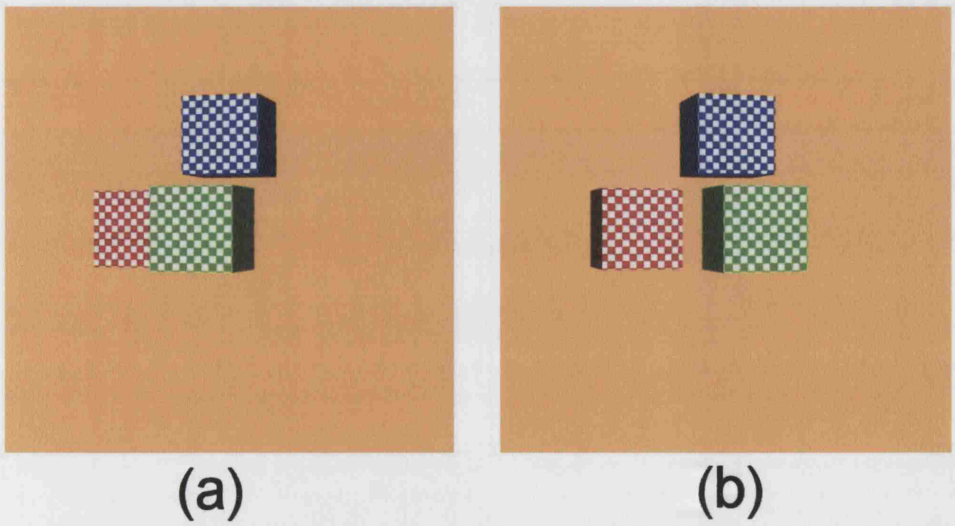


Figure 5.2. The sample basis views.

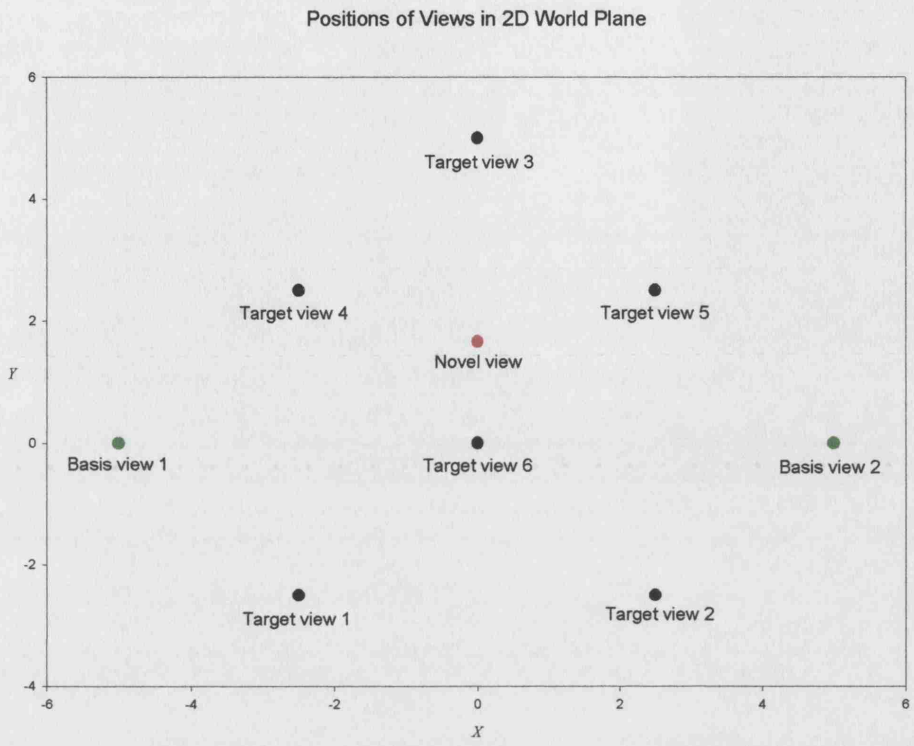


Figure 5.3. Positions of the sample view in the world plane.

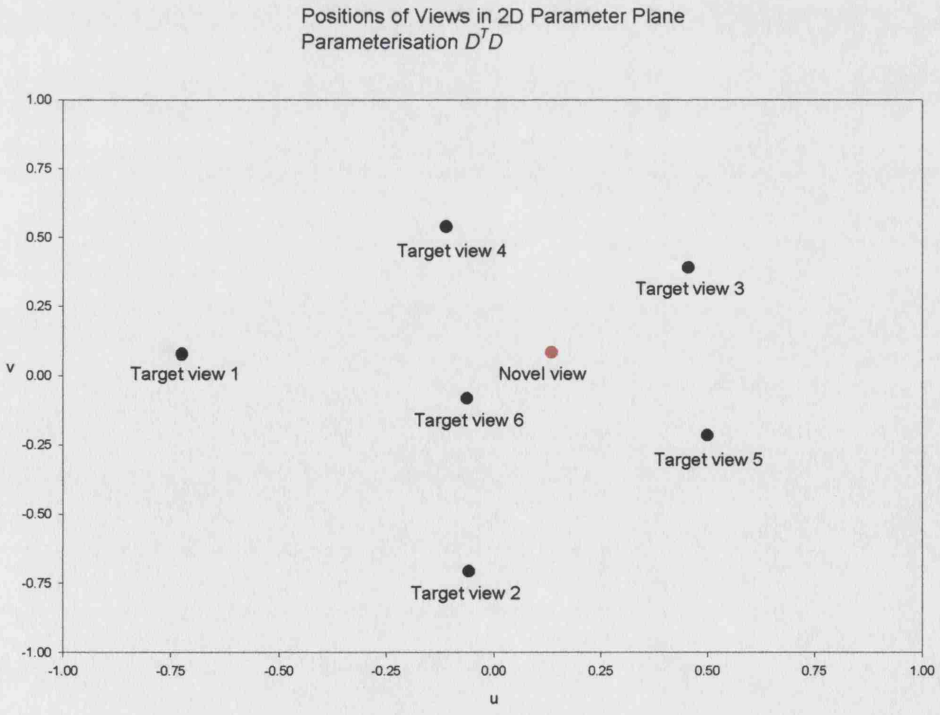


Figure 5.4. Positions of the target views in the (u,v) parameter plane for D^TD.

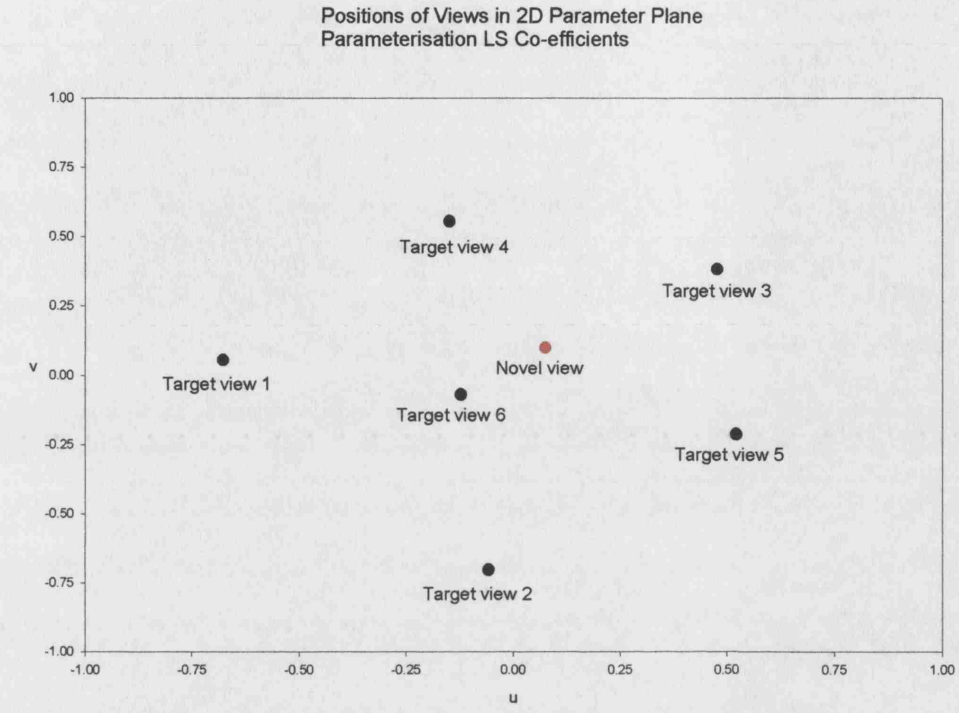


Figure 5.5. Positions of the target views in the (u,v) plane for the LS co-efficients.

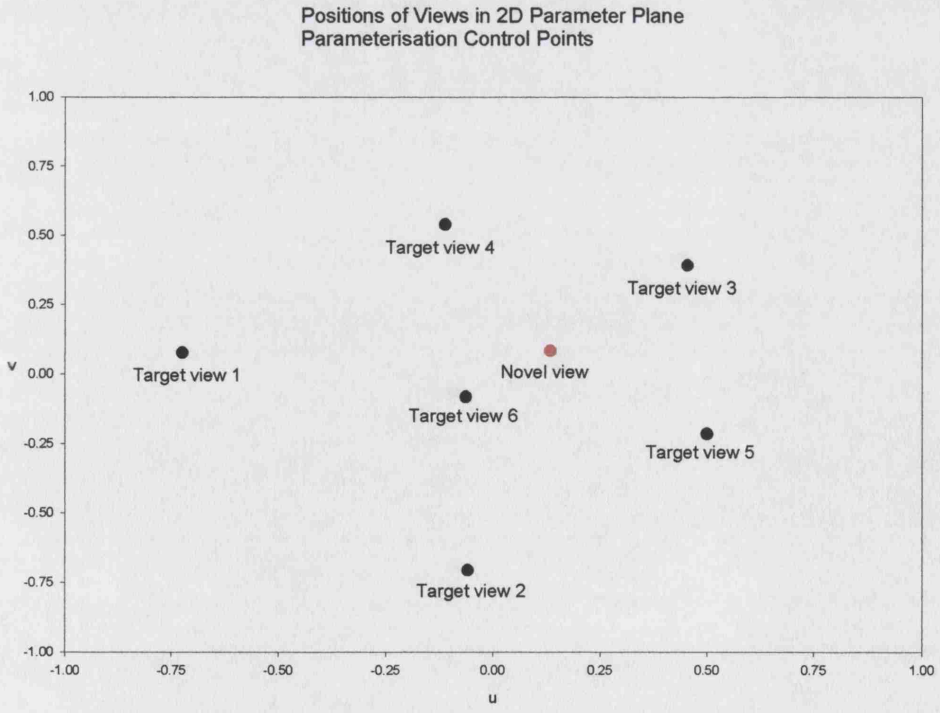


Figure 5.6. Positions of the target views in the (u,v) plane for the control points.

We also parameterised the six target views (figure 5.1) using the co-efficients of the LS relationships as described in section 5.4. Figure 5.5 shows the resulting positions of the six target views in the parameter plane. Again we can see that the relative positions of the target views are similar to the actual relative positions of the views in figure 5.3 and to the positions in figure 5.4.

Finally we parameterise the target views using the co-ordinates of the twenty-four control points in each of the six target views, as described in section 5.5. In this case we do not use any information from the basis views. The positions of the six views in the 2D parameter plane are shown in figure 5.6.

We can see that the relative positions of the six target views in figures 5.3 to 5.6 are all in a similar arrangement. The co-ordinate range in figure 5.3 is very different from those in figures 5.4 to 5.6. This is because the locations in figure 5.3 are the actual real world locations whereas 5.4 to 5.6 are the result of the parameterisation, where the points have been adjusted so that they satisfy the constraints in equation (5.2.5). The actual locations of the views are not important, as we are only interested in their relative positions. Figures 5.3 to 5.6 show that the method of parameterising the views produces sensible arrangement of the target views for each of the three different ways that we have chosen to parameterise. We will now use each of the three methods to synthesise a novel view and compare the accuracy of the three novel views.

In order to compare the accuracy of the novel views we need a reference image. Since we are using computer-generated images we are able to render an image with the camera at a known point and use each of the three parameterisations to synthesise that view. The view that we have chosen to synthesise is in the centre of the triangle of target views 4, 5 and 6 in figure 5.3. The position of this view is shown in red in figure 5.3. We do not know the exact corresponding position of this view in each of the 2D parameter planes in figures 5.4, 5.5 and 5.6. However we do know the exact position of each of the control points (x_i, y_i) in the view we are trying to construct. We can also use the point at the centre of the triangle defined by target views 4, 5 and 6 in each of figures 5.4, 5.5 and 5.6 as a good estimate of the position of the view we wish to synthesise. We can use this information as the starting point for finding the values of the parameters u and v in each of the three parameter planes

that allow us to produce an accurate, synthesised novel view in each of the three cases.

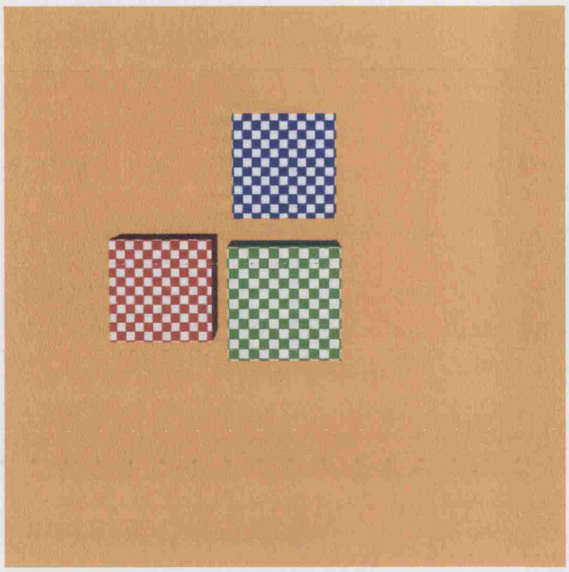
To synthesise a novel view we need to determine the positions of the control points in the view we are synthesising. In the case where we have parameterised the sample views in terms of the $D^T D$ matrix or the LS relationships we obtain a pair of multi-view relationships between the positions of the control points in the basis views and the novel view we are synthesising. These multi-view relationships can then be used to transfer the control points from the basis views into the novel target. In the case where we have parameterised the sample views in terms of the positions of the control points we obtain the positions of the control points as functions of the parameters u and v.

In order to synthesise the most accurate novel view in each of the three cases we wish to minimise the error between the exact known positions of the control points (x_i, y_i) and the control points $(\hat{x}_i, \hat{y}_i)$ that have been estimated from the parameterisation of the sample views. In each of the three cases we can use a minimisation process to find the values of parameters u and v such that the error value,

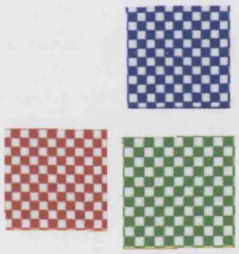
$$\varepsilon = \sum_{i=1}^{n} \varepsilon_i = \sum_{i=1}^{n} \left((\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 \right) \quad , \qquad (5.6.1)$$

is minimised. In each case the values of u and v at the centre of the triangle defined by target views 4, 5 and 6 were used as the starting point for the minimisation process. The minimisation process that we used was again Powell's method. The resulting values for u and v in the parameter planes are shown as red dots in figures 5.4, 5.5 and 5.6.

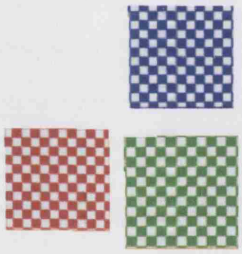
Once the positions of the control points in the novel view have been determined we can then render the intensities. Figure 5.7 shows the original sample view that we are trying to synthesise (part (a)) and four novel synthesised views ((b) to (e)). Part (b) in figure 5.7 has been synthesised by parameterising the elements of the matrix $D^T D$ and using the method described in section 4.4.2. Part (c) has been synthesised by using the co-efficients of the LS relationships to parameterise the sample views and using the method described in section 3.4.2 to render the intensities. The basis views in figure 5.2 have been used to synthesise these images.



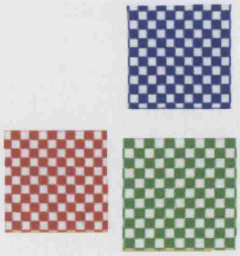
(a)



(b)



(c)



(d)

(e)

Figure 5.7. Synthesised novel views.

Figure 5.7 (d) and (e) shows two novel views that have been synthesised by using the positions of the control points as parameters. The two views in figure 5.2 have been used as the basis views to render figure 5.7 (d) and target views 5 and 6 (figure 5.1 (e) and (f)) have been used as the basis view to synthesise 5.7 (e).

It can be seen that the synthesised views (figure 5.7 (b) to (e)) all look accurate and are all very similar. To get a measure of how accurate each of the views are we can look at the error on the control points, i.e. the value of the error in equation 5.6.1. Table 5.8 shows the total sum of squared error on the control points, ε, and maximum error (maximum ε_i value) for each of the three parameterisations. It can be seen that the when we parameterise the co-efficients of the LS relationships we obtain lower error values than when we parameterise the matrix $D^T D$. This is what we would expect because the TLS relationships assume that there are errors on the control points in the basis views and we do not take these errors into account when using the TLS multi-view relationships to transfer points to the novel view. However, as we might expect, the lowest error values are obtained when we use the control points themselves to parameterise the sample views.

Method of Synthesising Sample Views	**Image in Figure 5.7**	**Total Sum Squared Errors on Control Points**	**Maximum Sum Squared Error**
$D^T D$	(b)	13.7759	2.24813
Co-efficients of LS relationships	(c)	12.9318	1.80165
Control points	(d) & (e)	1.0476	0.15222

Table 5.8. Errors on the control points of the novel views.

If we look at the synthesised novel views in figure 5.7 (b) to (e) we can see that in (b), (c) and (d) the top faces of the red and green cubes and the face on the right of the red cube are missing, although they can be seen in the actual view (figure 5.7 (a)). The reason that these faces are missing is that they cannot be seen in the basis views (figure 5.2) that were used to render the intensities. It was mentioned in section 5.5 that one of the advantages of parameterising the sample views by using the positions of the control points was that we do not need to keep the basis views fixed.

In figure 5.7 (e), where we have chosen two different views as the basis views, we are able to render the faces of the cubes that are missing from the other synthesised views.

5.6.2 Calibration Targets

In this section we will use each of the three methods described in sections 5.3 to 5.5 to parameterise a set of sample views of a pair of calibration targets. The sample images are 768×512 pixels and are shown in figure 5.9. The images were taken from various camera positions and using various focal length settings. The camera was angled so that it was pointing in the direction of the object and there was no rotation about the optical axis of the camera. There are thus four degrees of freedom associated with the camera, three for the translation and one for the focal length.

Although we know that there are in fact four degrees of freedom we continue to use only two parameters to parameterise the sample views. In order to parameterise the sample views by using four parameters we would require a minimum of 15 sample target views. When we parameterise the data matrix $D^T D$ and the co-efficients of the LS relationships we use the images (a) and (b) in figure 5.9 as the basis views and (c) to (h) as the six target views. When we parameterise the control points we use images (c) to (h) in figure 5.9 as the sample views. The control points were taken to be the six corners of the calibration targets and the tip of Sleepy's beard in the Snow White picture.

We do not show the positions of the six views in the 2D parameter planes as we know that the sample views have four degrees of freedom and so cannot easily make a comparison between the two. Instead we use the results of the parameterisations to synthesise novel views. We have chosen to synthesise the views lying at the centre of the triangle formed by images (c), (f) and (h) in figure 5.9 in the three different parameter planes. The three synthesised images are shown in figure 5.10. Part (a) in figure 5.10 shows the image synthesised when the sample views are parameterised using the elements of $D^T D$. Part (b) shows the synthesised image when the sample views have been parameterised using the co-efficients of the LS relationships. Finally, figure 5.10 (c) shows the synthesised image when the parameterisation is performed using the control points.

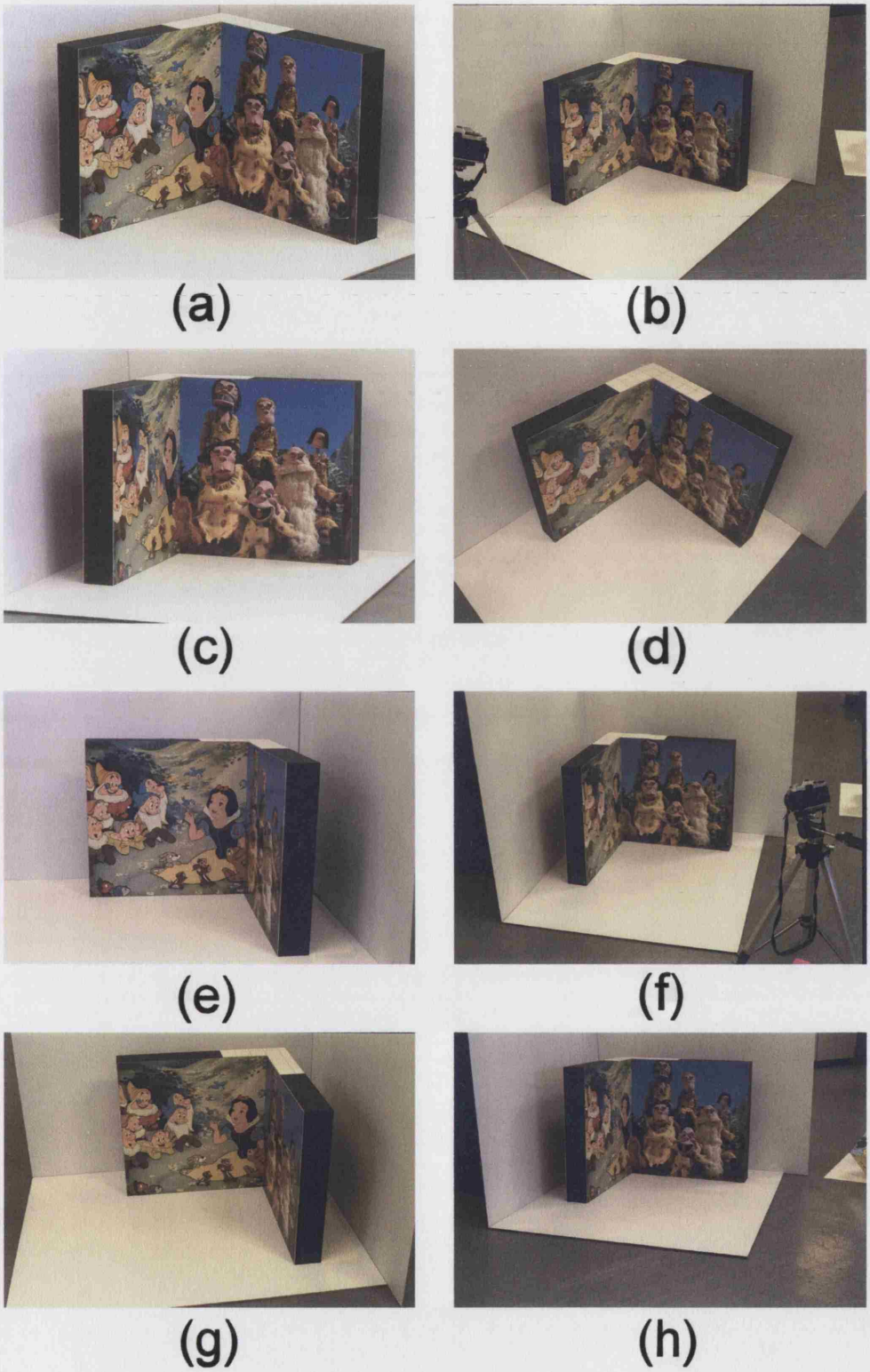


Figure 5.9. Sample views of the calibration targets.

If we look at the synthesised images in figure 5.10 we can see that the images are slightly blurred but realistic in appearance. The blurring is due to the interpolation of the intensities in the two basis views. Since we are using a set of sample images taken at unknown camera positions we do not have a target image to compare with the synthesised images in figure 5.10. However, if we look at the overall appearance of the synthesised images (figure 5.10) it is possible to say that (c) looks the most realistic. The ratios of parallel lines are invariant under affine imaging conditions [Zis92] and if the parallel lines are the edges of a square this ratio is equal to one. If we look at the right calibration target in each of images (c), (f) and (h) the images in figure 5.9 and calculate the ratio of the length of the right vertical edge to the length of the left vertical edge we obtain values of 0.97, 0.97 and 0.96 respectively. These values are close to the value one and indicate that images (c), (f) and (h) have been obtained under approximately affine conditions. Since these sample images are approximately affine we would also expect the novel view to be approximately affine. If we now calculate the values of the same ratio in the novel views in figure 5.10 we obtain values of 0.80, 0.88 and 0.99 in (a), (b) and (c) respectively. We can see that the value of the ratio of the lengths of the edges in the novel view figure 5.10 (c) is closer to the value of the same ratio in the sample views and closer to the value of one than the two other novel views.

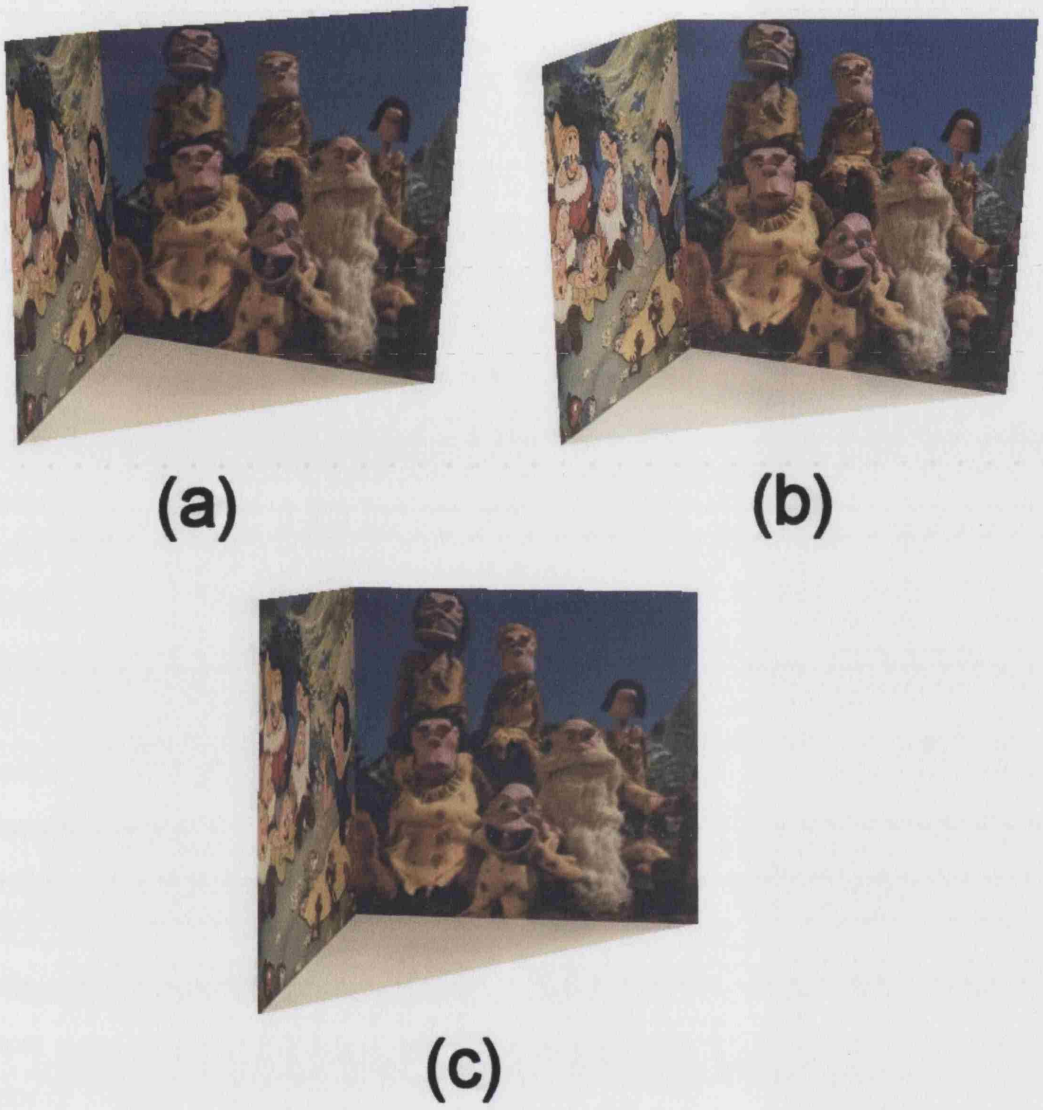


Figure 5.10. Novel views of the calibration targets, generated by parameterising the co-efficients of the LS relationships in (a), the elements of $D^T D$ in (b) and the co-ordinates of the control points in (c).

5.6.3 Face Images

In the previous two sections we have used each of the three methods described in sections 5.3 to 5.5 to parameterise two sets of sample images and then to synthesise novel views. We have seen that parameterising the views in terms of the co-ordinates of the control points has advantages over the other two methods. In this section we use the method of parameterising the control points to characterise a set of sample views of a face and use the results to synthesise novel views. The six sample views are 1024×1536 pixels and are shown in figure 5.11. The images were taken using a camera mounted on a vertical metal rod as shown in figure 5.12. Images were taken with the camera at several different distances along the vertical rod. The height of the camera was measured for each camera position. The vertical rod was then moved in a horizontal direction with the camera at a fixed height on the vertical rod. Again we recorded the horizontal distance that the camera was moved. Since the images are of a person and they were not taken simultaneously it is possible that the scene may have changed slightly between images. We could avoid this occurring be using six cameras to take the images simultaneously. At each camera position the camera was rotated such that it pointed directly at the face of the person being photographed. The camera was not rotated about its optical axis. Therefore, in this example we can assume that there are two degrees of freedom for the position and orientation of the camera. The approximate positions of the camera used to obtain the six sample views in figure 5.11 are shown in a plane in figure 5.13.

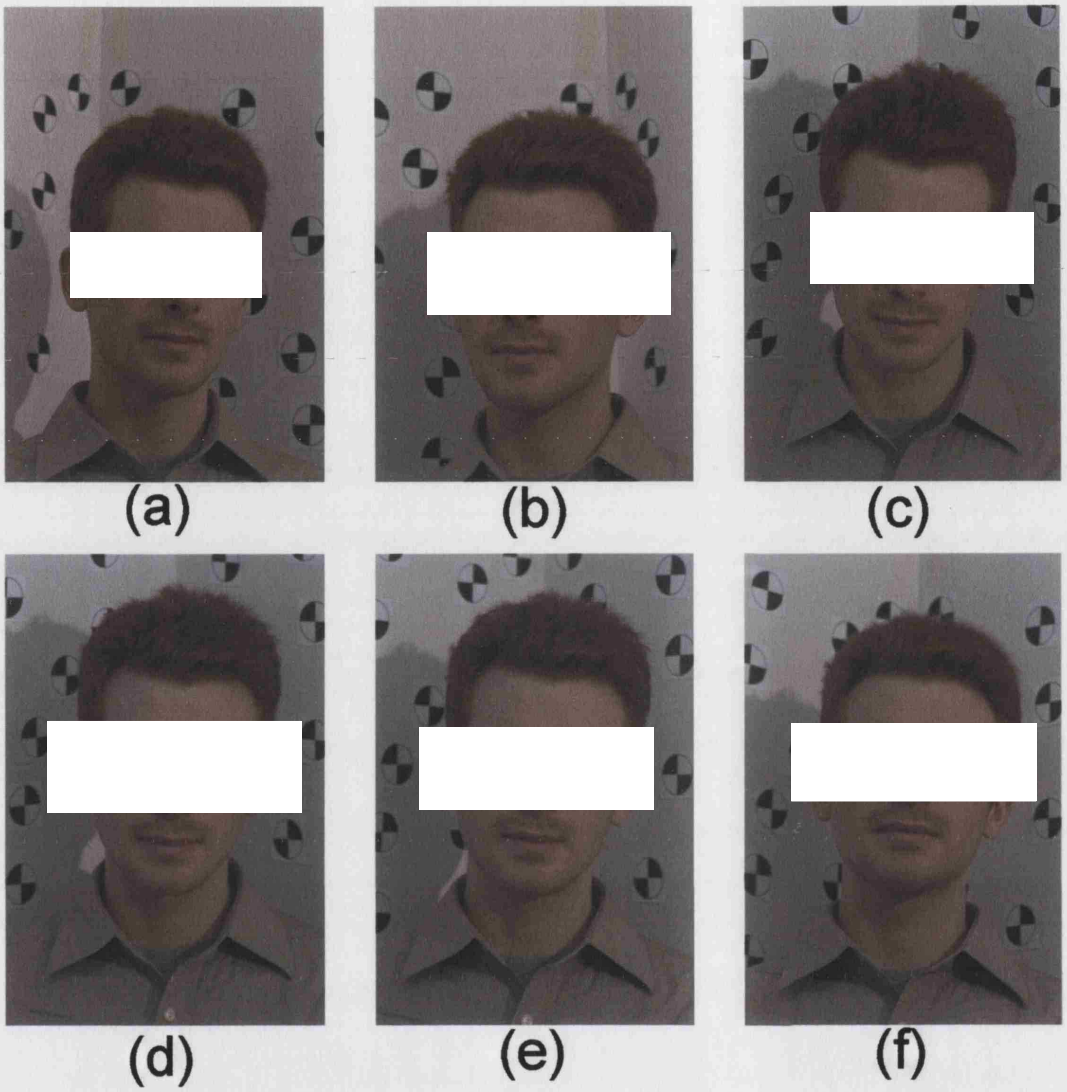


Figure 5.11. Sample views of face images.

The sample images were then parameterised using the positions of a set of 43 control points located on the face and the results used to generate four novel views of part of the face. The positions of the six sample views in the parameter plane are shown in figure 5.14 along with the positions of the four synthesised novel views. The four novel views are shown in figure 5.15. It can be seen that all the synthesised images look realistic. If we look at the positions of the sample views in the parameter plane in figure 5.14 and compare them with the actual positions of the cameras in space in figure 5.13 we can see that they do not correspond particularly well. However, although the positioning of the sample views in the parameter plane is not accurate we can still use the results to interpolate between the images and obtain realistic synthesised novel views.

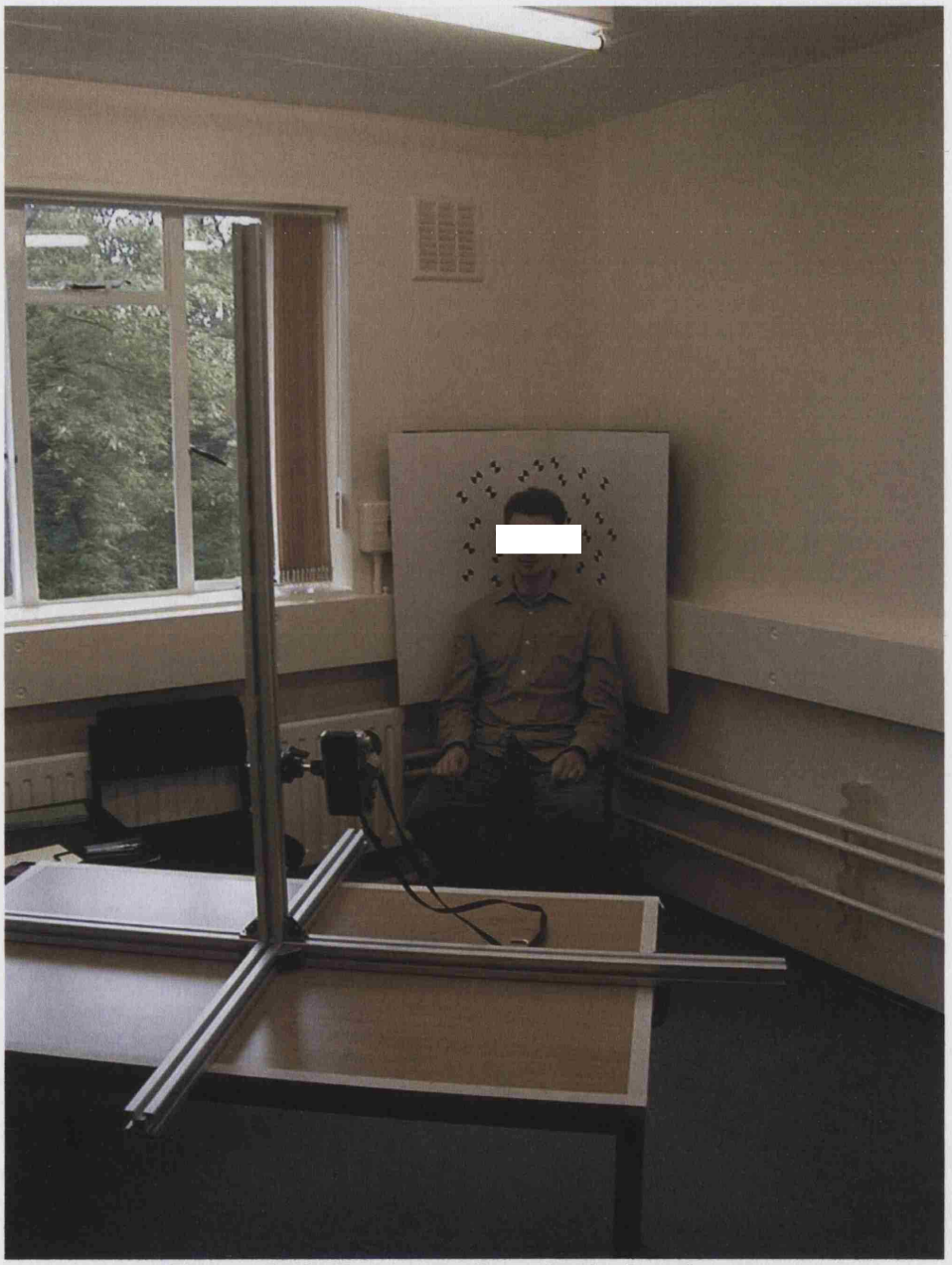


Figure 5.12. Set-up used to obtain face images.

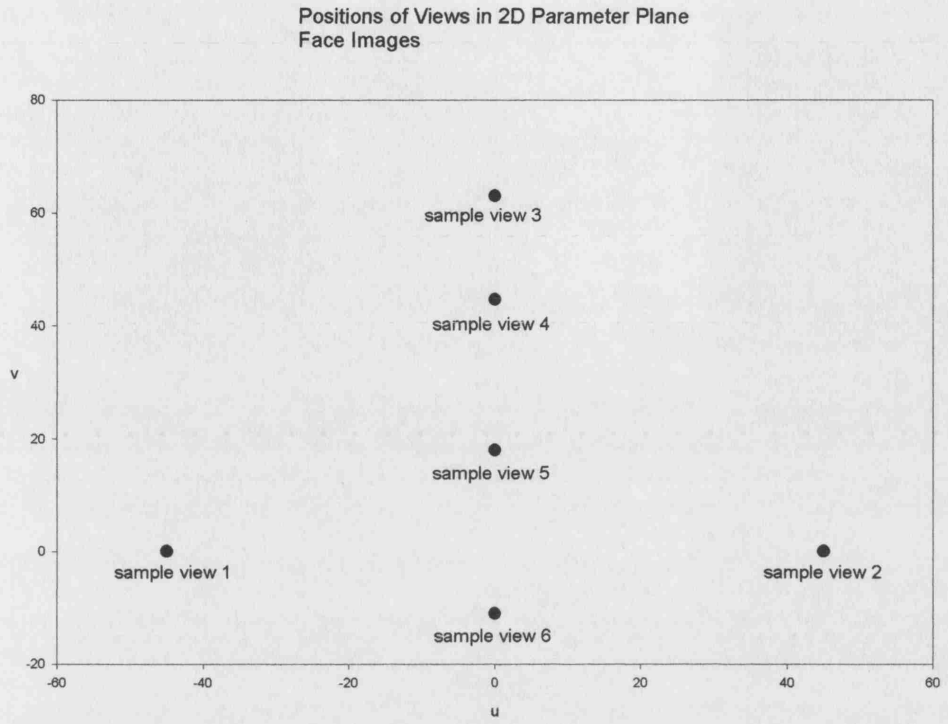


Figure 5.13. Positions of sample views in the world plane.

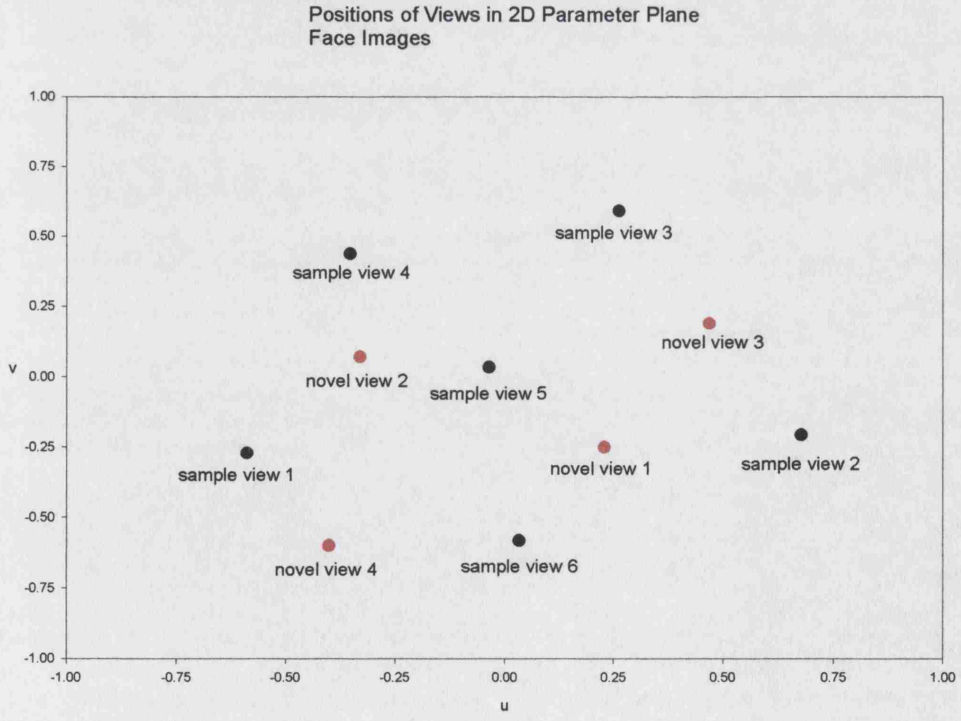


Figure 5.14. Positions of the sample views and novel views in the (u,v) plane.

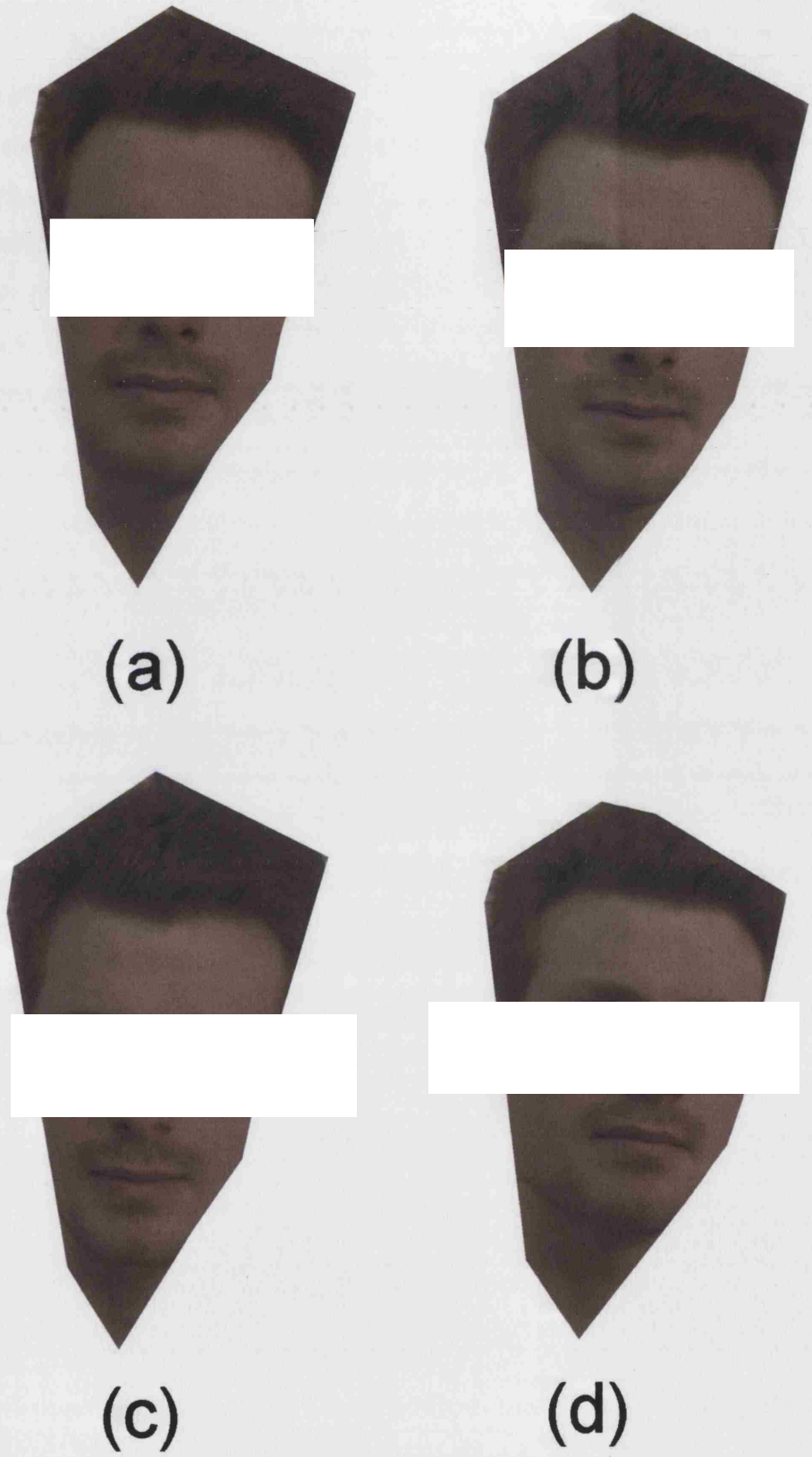


Figure 5.15. Novel views of the face images.

5.7 Conclusions

In this chapter we have described a method of synthesising novel views using only a set of sample views and information that can be extracted directly from those views. In order to construct a novel view we first need to parameterise the sample views. The parameterisation is described in section 5.2 and can be used to characterise the data matrix $D^T D$, the LS relationships or the control points. The results of the parameterisation can then be used to interpolate between the views and determine the positions of a set of control points in a novel view. We have seen that parameterising the control points has several advantages over the other two methods. Only six sample views are needed to parameterise the control points whereas the other two methods require eight views. Another advantage of parameterising the control points is that we do not need to keep the basis views fixed. When synthesising a novel view we are able to choose any of the sample views to use to render the intensities in the novel view. As well as these advantages, parameterising the control points leads to a more accurate and more realistic synthesised view as we have seen in sections 5.6.1 and 5.6.2.

In section 5.6.3 we have shown that even when the parameterisation does not position the sample views in an arrangement very similar to the actual camera viewing positions we are still able to generate realistic novel views.

Chapter 6

A Limit of Extrapolation?

In the previous chapter we described a method of synthesising novel views starting from a set of sample images. The method parameterises a set of sample views in terms of two parameters u and v. We can vary these values of the parameters to interpolate between the sample views. As well as being able to interpolate between the views, it is also possible to extrapolate away from the sample views and generate novel views that are outside the original set of sample views.

When extrapolating away from the sample views we would expect the quality of the synthesised views to deteriorate as the position of the novel view is moved away from the sample views. In this chapter we explore the structure within the total least squares solution to try and determine whether it is possible to estimate a limit of extrapolation.

In sections 6.1 and 6.1.1 we give an example of how the TLS relationships can breakdown when used on images of a symmetric object. In section 6.1.2 we describe how the TLS relationships change as we move away from the case of a symmetric object to a more general configuration of world points. In section 6.2 we use the theory from chapter 5 to locate the point where the TLS relationships breakdown for the case of the symmetric object and to predict the variation of the singular values in asymmetrical examples. We also show that it is possible to predict the point where the TLS relationships breakdown using images of a real symmetric object. Finally, conclusions and some suggestions for further work are given in section 6.3.

6.1 Breakdown of The Multi-View Relationships

Recall the form of the TLS relationships from equation (5.3.1):

$$\begin{aligned} l_1x_i + l_2y_i + l_3x_i' + l_4y_i' + l_5x_i'' + l_6y_i'' = 0 \\ m_1x_i + m_2y_i + m_3x_i' + m_4y_i' + m_5x_i'' + m_6y_i'' = 0 \end{aligned} \qquad (6.1.1)$$

As usual in equation (6.1.1) we assume that the control points are expressed using centre of mass co-ordinates.

Suppose that we wish to render a target view that has been encoded as a linear combination of a pair of basis views using the TLS relationships. The first step is to recover the co-ordinates of the control points in the target view. However, if the relationships in (6.1.1) cannot be used to solve for the target view co-ordinates, (x_i, y_i), we cannot render the target view and call this a breakdown of the TLS relationships.

In general the relationships cannot be solved when the ratios l_1/l_2 and m_1/m_2 are equal, or equivalently when the value of $l_1m_2 - l_2m_1$ is equal to zero. In this chapter we give an example of when this occurs and show how the value $l_1m_2 - l_2m_1$ can be used to determine a limit of extrapolation. In general the value of $l_1m_2 - l_2m_1$ will be non-zero and we will not be able to use this value to determine a limit of extrapolation in the general case. Some suggestions of how to determine a limit of extrapolation in the general case are given at the end of this chapter, in section 6.3.

One particular example of when the pair of relationships in (6.1.1) cannot be used to solve for the target view co-ordinates is when $l_2 = m_2 = 0$ (or similarly $l_1 = m_1 = 0$). In this case the y coefficient (or x coefficient) in both equations is equal to zero and therefore we cannot recover the y co-ordinates (or x co-ordinates) of the points in the target view. Obviously when $l_2 = m_2 = 0$ the value of $l_1m_2 - l_2m_1$ will also be equal to zero. A particular example of when the values of l_2 and m_2 can both become equal to zero is in the case where there is a high level of symmetry in both the configuration of the world points and in the positioning of the cameras. An example of this is described in section 6.1.1.

In section 6.1.2 we will show that as we move away from the symmetric case the values of l_2 and m_2 will no longer be equal to zero but will still be small in

comparison to the other co-efficients. Consequently we may be able to solve for the x and y co-ordinates but they may be more likely to be affected by errors. In sections 6.1.1 and 6.1.2 we explore what happens to the value of $l_1 m_2 - l_2 m_1$ as we extrapolate away from the basis views in the symmetric and near symmetric cases.

6.1.1 The Symmetric Case

We will now describe a specific set-up where the TLS relationships breakdown. The object being imaged is a set of six cubes arranged in a symmetric configuration about its X, Y and Z axes. The object has its centroid at $(0,0,0)$. Each cube has sides of length 1.4 and the cubes are centred at 1.7 units along each of the positive and negative X, Y and Z axes. The control points are taken to be the 48 vertices of the cubes. The basis view cameras are placed at $(5,0,18)$ and $(-5,0,18)$, and the target view camera is initially placed mid-way between the two basis view cameras at $(0,0,18)$. All cameras have the optical axis passing through the centroid of the object. The set-up described here is illustrated in figure 6.1. The target view camera is then moved away from the baseline in the direction of Y_v, which is the direction of the positive Y axis. Since we are using synthetically generated images we can use all the control points in each of the views even though they may not all be visible in all the images.

The symmetry within the configuration of object points and the arrangement of the cameras provides us with some additional relationships between the corresponding points. Not all of the relationships are necessary for what we are about to show. The symmetry that is required is that the target view is symmetrical about the y axis and the basis views are reflections of each other in the y axis as shown in figure 6.2. If there is a control point at (x,y) in the target image corresponding to points $(x',y')=(a,b)$ and $(x'',y'')=(c,d)$ in the two basis views then there will also be a control point at $(-x,y)$ in the target image which corresponds to points $(x',y')=(-c,d)$ and $(x'',y'')=(-a,b)$ in the two basis views.

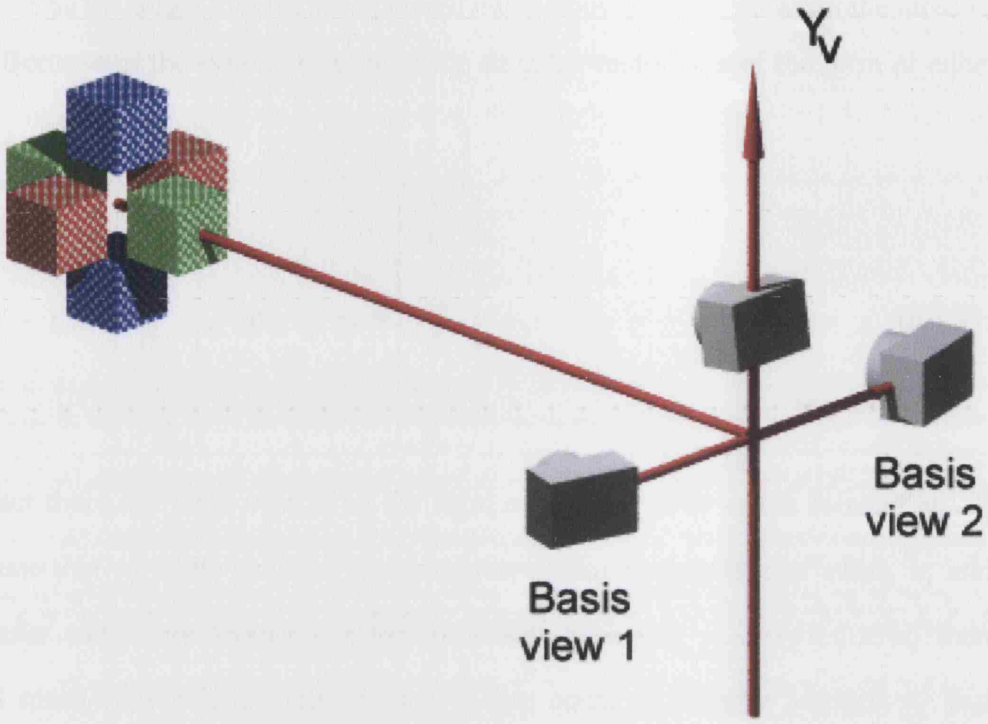


Figure 6.1. Arrangement of the camera positions relative to the symmetrical object.

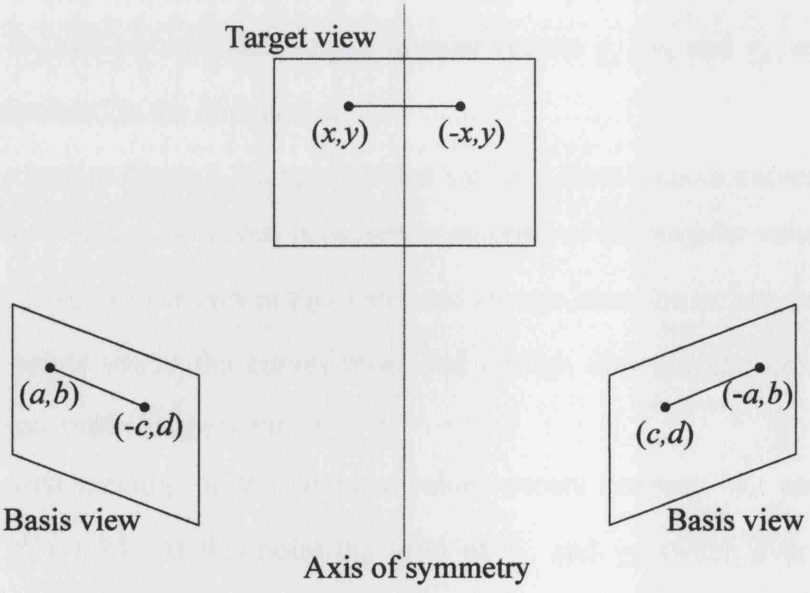


Figure 6.2. Positions of a pair of symmetrical control points in the three views

We will now look at what happens to the singular values and the singular vectors as the target view is extrapolated away from the basis views in the direction of Y_v. Because of the symmetry each of the singular vectors are of the form of either $\underline{v}_a$ or $\underline{v}_b$, where

$$\underline{v}_a = \begin{pmatrix} a_a \\ 0 \\ c_a \\ d_a \\ c_a \\ -d_a \end{pmatrix}, \text{ and } \quad \underline{v}_b = \begin{pmatrix} 0 \\ b_b \\ c_b \\ d_b \\ -c_b \\ d_b \end{pmatrix} . \qquad (6.1.2)$$

In fact there are three vectors of the form of $\underline{v}_a$ and three of the form of $\underline{v}_b$. If we assume that $\underline{v}_i$ is the singular vector corresponding to the singular value w_i and the singular values are arranged in the usual way such that $w_1 \geq w_2 \geq \ldots \geq w_6$ then the TLS multi-view relationships $\underline{l}$ and $\underline{m}$ are equal to singular vectors $\underline{v}_6$ and $\underline{v}_5$ respectively. As long as we have one of $\underline{v}_6$ and $\underline{v}_5$ of the form of $\underline{v}_a$ and one of the form of $\underline{v}_b$ then we can solve for the target view co-ordinates.

As we begin to extrapolate the target view along Y_v we find that $\underline{l}$ and $\underline{m}$ ($\underline{v}_6$ and $\underline{v}_5$) have different forms and it is possible to solve for the target view co-efficients x and y. Figure 6.3 shows what happens to the three smallest singular values, w_6, w_5 and w_4 corresponding to singular vectors $\underline{v}_6$, $\underline{v}_5$ and $\underline{v}_4$, as the target view is extrapolated in the direction of Y_v.

If we look at figure 6.3 it appears that we have three smooth curves that cross at two points along Y_v. However, because the ordering of the singular values requires that $w_4 \geq w_5 \geq w_6$ we get curves that meet and change direction at two points along Y_v. At the points where the curves meet and change direction the corresponding singular vectors may change form.

The first meeting of the singular values occurs between w_6 and w_5 at a distance of $Y_v \approx 0.85$. At this point the form of $\underline{v}_6$ and $\underline{v}_5$ switch over. The next meeting occurs between w_5 and w_4 at a distance of $Y_v \approx 3.7$, at which point the form

of $\underline{v}_5$ and $\underline{v}_4$ switch over. Table 6.4 shows the how the form of the singular values changes as the target view is extrapolated along Y_v.

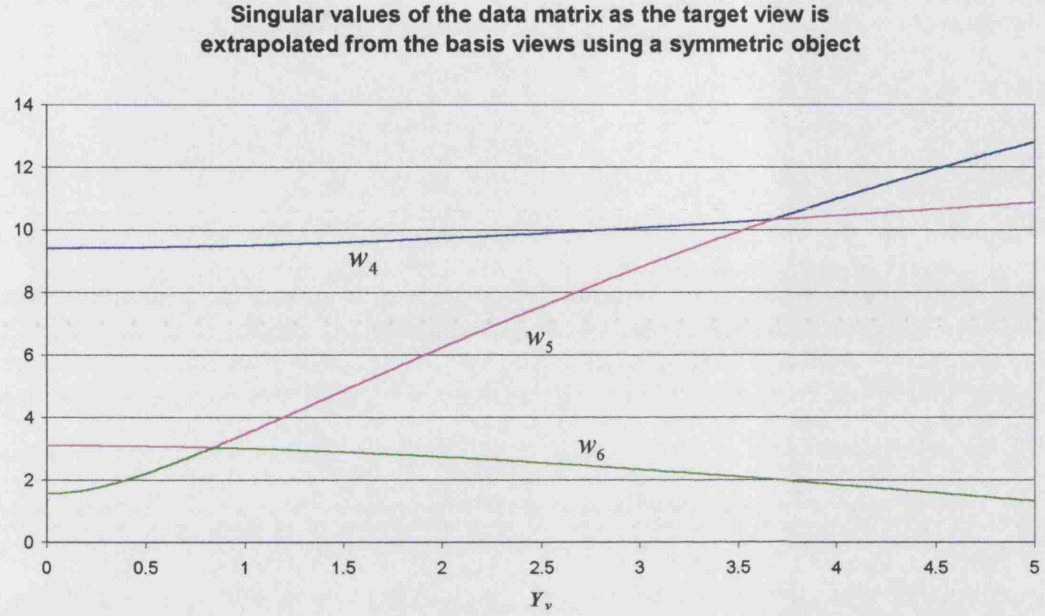


Figure 6.3. The three smallest singular values for the symmetrical object.

We can see from table 6.4 that the first meeting of the singular values at $Y_v \approx 0.85$ does not affect the multi-view relationships because the two smallest singular vectors have different forms and have merely been interchanged. However at the second crossing, $Y_v \approx 3.7$, the singular vectors $\underline{v}_5$ and $\underline{v}_4$ switch over and after this point both the vectors $\underline{v}_6$ and $\underline{v}_5$ are of the form of $\underline{v}_a$. At this point we can no longer solve for the y co-ordinates in the target view and the view synthesis/encoding procedure breaks down. We will call this point the critical viewpoint.

Range of Y_v	**Form of singular vector**		
	$\underline{v}_4$	$\underline{v}_5$	$\underline{v}_6$
$0 \le Y_v \le 0.85$	$\underline{v}_a$	$\underline{v}_a$	$\underline{v}_b$
$0.85 < Y_v \le 3.7$	$\underline{v}_a$	$\underline{v}_b$	$\underline{v}_a$
$3.7 < Y_v$	$\underline{v}_b$	$\underline{v}_a$	$\underline{v}_a$

Table 6.4. The form of the three smallest singular vectors over the range of Y_v.

It is also possible to determine the position of the critical viewpoint by looking at the absolute value of $l_1 m_2 - l_2 m_1$. Figure 6.5 shows the absolute value (because the signs of $\underline{l}$ and $\underline{m}$ are indeterminate) of $l_1 m_2 - l_2 m_1$ as the target view is moved along Y_v. We can see that at $Y_v \approx 3.7$ the value of $l_1 m_2 - l_2 m_1$ drops to zero and remains at zero for greater Y_v. Beyond this critical viewpoint, it is no longer possible to solve for the co-ordinates of the control points (x, y) in the target view. The symmetric example we have used here is a specific case where the view synthesis procedure breaks down. In general we will not have the level of symmetry in the images required for a critical viewpoint to exist and there will not be a specific point where the view synthesis procedure breaks down. It is important to note that the symmetry within the images means that it is possible for a critical viewpoint to exist. It does not necessarily mean that a critical viewpoint has to exist.

We will now explore what happens to the singular values and the singular vectors when we remove the symmetry.

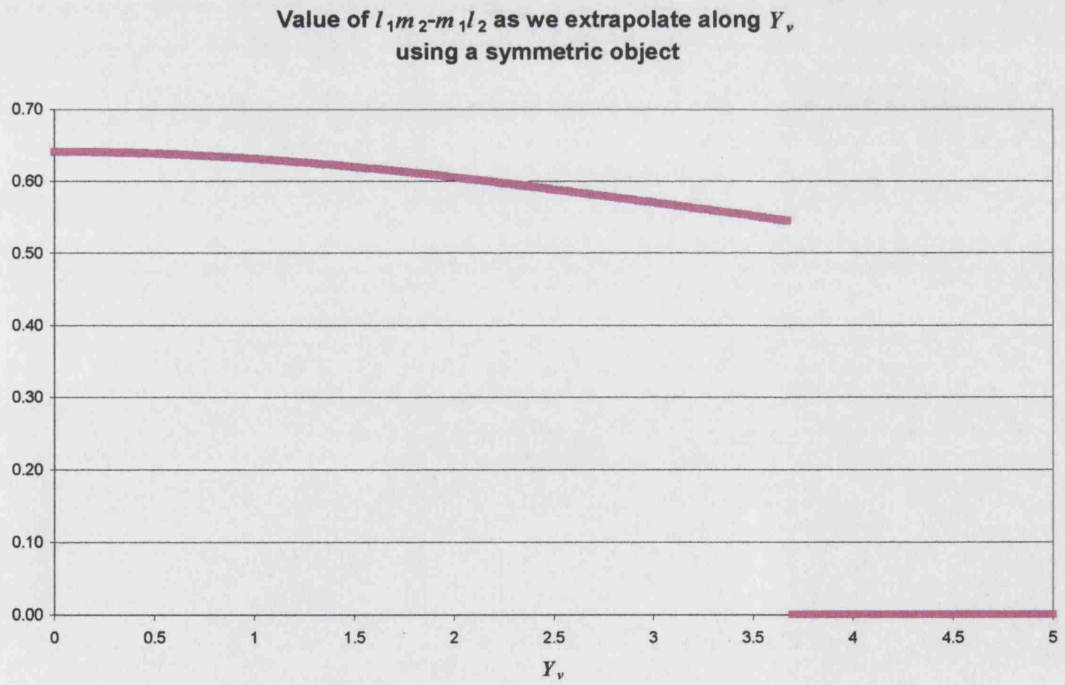


Figure 6.5. Absolute value of $l_1 m_2 - m_1 l_2$ for the symmetrical object.

6.1.2 The Asymmetric Case

In the previous section we gave an example of the TLS relationships breaking down. We will now look at what happens as we move away from the symmetric case. To do this we keep the camera positions and the direction of extrapolation the same as in the previous example. In order to perturb the symmetry of the object points we add a random error between -0.3 and 0.3 to each of the X, Y and Z co-ordinates of each of the object points. In this case the object will not be symmetrical but will be close to the symmetrical case.

Figure 6.6 shows the three smallest singular values as the target view camera is moved along Y_v. We can see that the two smallest singular values, w_6 and w_5, appear to meet at $Y_v \approx 0.8$. In fact the two curves do not actually meet, they get very close together (but do not touch) and change direction. As we move further along Y_v the two singular values w_5 and w_4 get closer together. Again, they do not touch as in the previous example, instead they continue to get close together until $Y_v \approx 4.05$ at which point the shape of the two curves changes and the distance between the curves starts to increase.

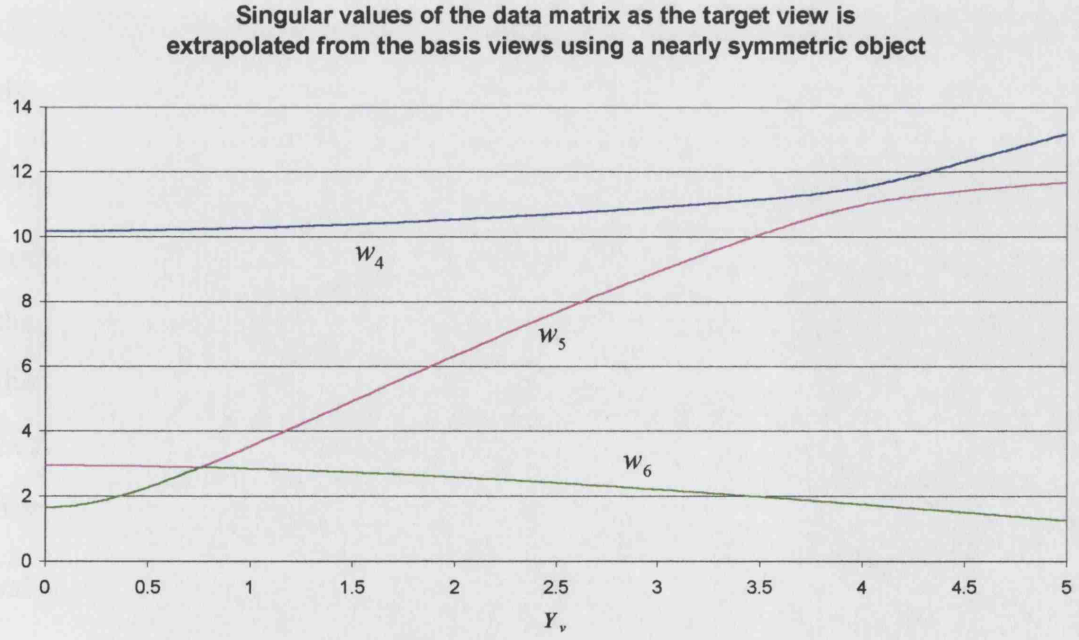


Figure 6.6. The three smallest singular values for the nearly symmetrical object.

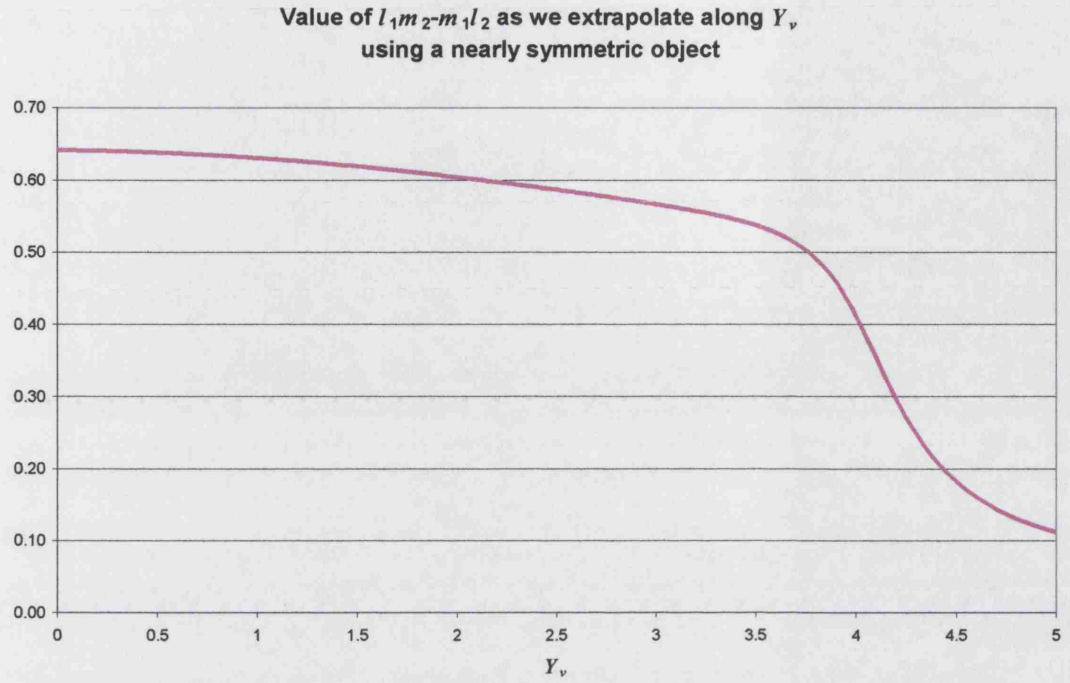


Figure 6.7. Absolute value of $l_1m_2 - m_1l_2$ for the nearly symmetrical object.

Figure 6.7 shows the absolute value of $l_1m_2 - l_2m_1$ for this near symmetric object as we extrapolate along Y_v. We can see that when $Y_v \approx 4.05$ the value of $l_1m_2 - l_2m_1$ starts to decrease rapidly. The values of l_2 and m_2 do not drop to zero for values of $Y_v \geq 4.05$ as in the symmetric case but they are both small in comparison to the other co-efficients. This is shown in figure 6.8, which plots the ratio $\left(l_2^2 + m_2^2\right) \Big/ \sum_{i=1,i\neq 2}^{6} \left(l_i^2 + m_i^2\right)$. In this case the TLS relationships do not break down completely and we can still solve for the x and y co-ordinates. However since both the l_2 and m_2 are small in comparison with the other coefficients we may expect that the y co-ordinate becomes more sensitive to any errors in the relationships as we extrapolate beyond $Y_v = 4.05$ than for values of $Y_v < 4.05$. In this case we do not have a critical viewpoint where the view synthesis breaks down but we can use the value of $l_1m_2 - l_2m_1$ or $\left(l_2^2 + m_2^2\right) \Big/ \sum_{i=1,i\neq 2}^{6} \left(l_i^2 + m_i^2\right)$ to give a limit of extrapolation.

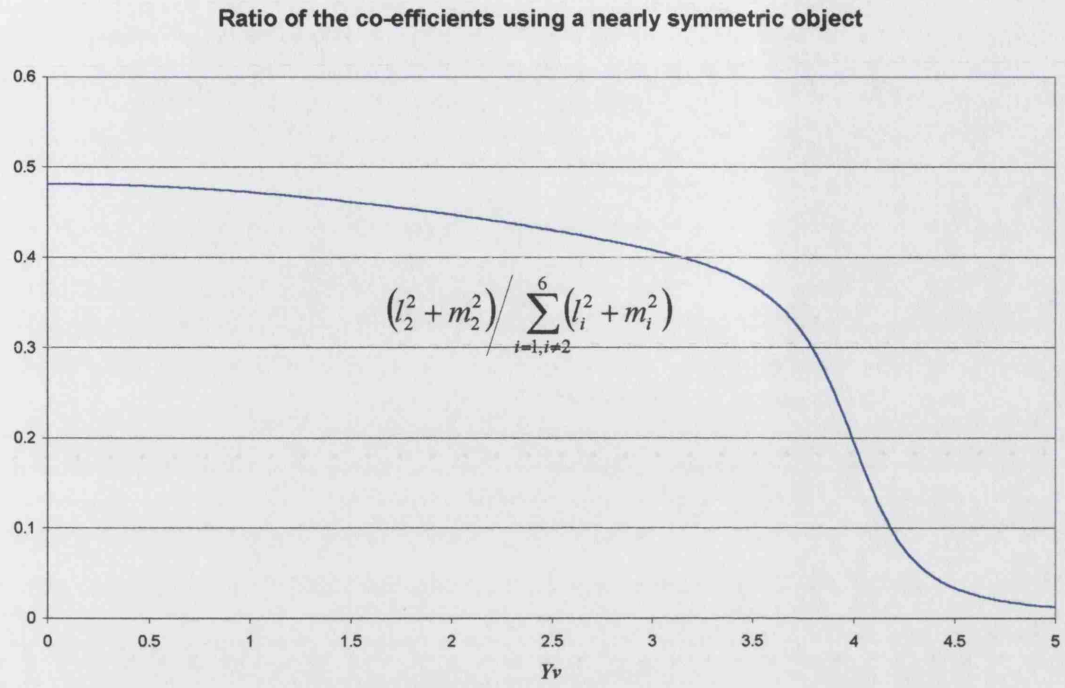


Figure 6.8. The ratio of $\left(l_1^2+m_1^2\right)\Big/\sum_{i=1,i\neq 2}^{6}\left(l_i^2+m_i^2\right)$ for the nearly symmetrical object.

In the previous test we have used an object which is close to being symmetrical. Although it does not have the same high level of symmetry as the object in section 6.1.1 it is still not a natural arrangement of world points. We will now repeat the experiment using a random object. In this example the cameras are arranged as in the previous two examples. Our object is generated using a set of 48 random points with each of the X, Y and Z co-ordinates for each of the points being within the range -2.5 to 2.5. The positions of the random points in the target image at $Y_v = 0$ are shown in figure 6.9. Figure 6.10 shows the three smallest singular values and figure 6.11 shows the absolute value of $l_1 m_2 - l_2 m_1$ as we extrapolate along Y_v.

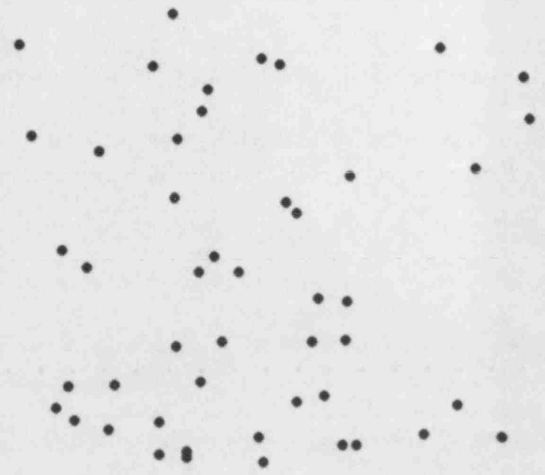


Figure 6.9. Image of the random set of control points at $Y_v = 0$

We can see from figure 6.10 that the two smallest singular values, w_6 and w_5, no longer cross. We can also see that the distance between w_5 and w_4 decreases as we extrapolate along Y_v. The curve of the singular value w_6 is decreasing over the range $1.8 \leq Y_v \leq 7.2$ until a minimum value is reached at $Y_v \approx 7.3$. After this point the value w_6 starts to increase again. This is a similar result to that seen in section 4.3.3 and in figures 4.8 and 4.9. Although we do not know what causes this decrease in the singular values we have chosen not to investigate this further in this thesis. In this experiment we have extrapolated out to a distance of $Y_v = 20$, which is four times further than in the previous examples of the symmetrical and nearly symmetrical objects. There is no obvious point along the curves which indicates that a breakdown of the total least squares solution should occur.

If we look at the absolute value of $l_1 m_2 - l_2 m_1$ in figure 6.11 we can see that it is initially decreasing as we extrapolate along Y_v. A minimum value for $l_1 m_2 - l_2 m_1$ is reached at a distance $Y_v \approx 13.1$ and then the value of $l_1 m_2 - l_2 m_1$ starts to increase. In this case a critical viewpoint does not exist and it is not possible to use the singular values or the value of $l_1 m_2 - l_2 m_1$ to determine an obvious limit of extrapolation.

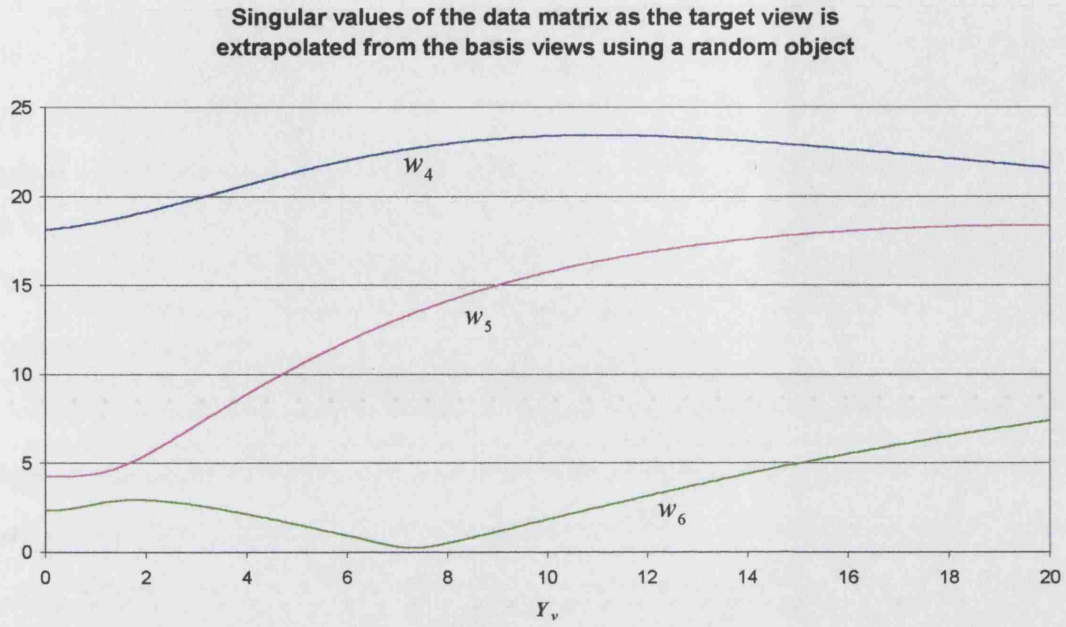


Figure 6.10. The three smallest singular values for the random object.

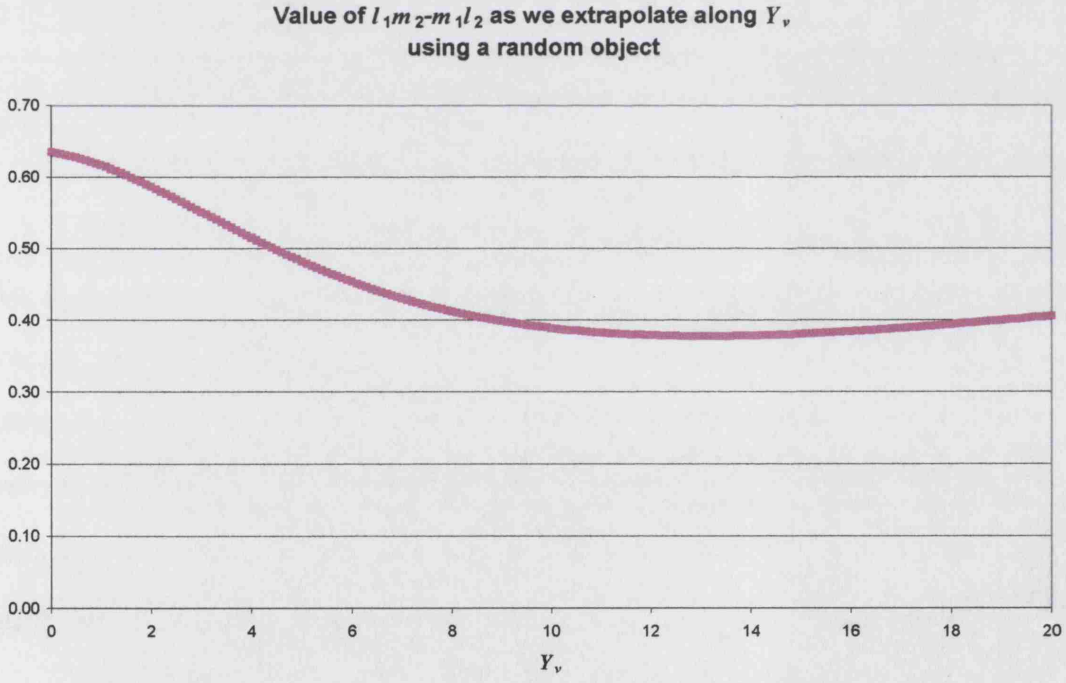


Figure 6.11. Absolute value of $l_1m_2 - m_1l_2$ for the random object.

6.2 Evaluation

In this section we use the theory from chapter 5 and the objects described in section 6.1 to determine how accurately the parameterisation can predict the shape of the

curves of the singular values and, in the case of the symmetrical object, the position of the critical viewpoint. We use the same camera positions as in 6.1.1 and the direction of extrapolation is along Y_v. Since we are only moving the target view camera along a line we only need to include one parameter in the fitting process. Therefore we can adapt the theory from chapter 5 to only include one parameter in the fitting process. We will now explain the theory as it will be used for the evaluation in sections 6.2.1 to 6.2.3. We can write each control point x and y as a function of one parameter u, instead of two as in equation (5.2.2) which becomes:

$$E_{ji} = \alpha_j u_i^2 + \beta_j u_i + \gamma_j \quad . \tag{6.2.1}$$

The E_j are the control point co-ordinates x and y, $i = 1..m$ where m is the number of sample views and $j = 1..2n$ where n is the number of control points. Since we are using only one parameter we need only three sample target views to find a solution for u_i and α_j, β_j and γ_j. We find a solution using the minimisation process described in section 5.2. In this case the value S (in (5.2.5)) that we want to minimise becomes:

$$S = \sum_{j=1}^{2n} \alpha_j^2 \quad , \tag{6.2.2}$$

and our set of constraints equivalent to (5.2.6) are:

$$\sum_{i=1}^{6} u_i^2 = 1, \quad \text{and} \quad \sum_{i=1}^{6} u_i = 0 \quad . \tag{6.2.3}$$

To begin the minimisation process we set $\tilde{u}_1 = 0$ and $\tilde{u}_2 = 1$, and assign arbitrary values to the remaining parameters $\tilde{u}_3$ to $\tilde{u}_m$. These parameter values are then transformed into the values u_i such that the conditions in (6.2.3) are satisfied. The values of u_i can then be used to calculate the co-efficients α_j, β_j and γ_j. By varying the parameters $\tilde{u}_3$ to $\tilde{u}_m$ we can find a solution for u_i and α_j, β_j and γ_j such that S in (6.2.2) is minimised.

To begin this experiment we use three target views at $Y_v = 0$, $Y_v = 1$ and $Y_v = 2$ and set the three respective parameter values to $\tilde{u}_1 = 0$, $\tilde{u}_2 = 1$ and $\tilde{u}_3 = \rho$, where ρ is an arbitrary value. By varying the value of ρ we find a solution for u_1, u_2 and u_3, and each of the α_j, β_j and γ_j.

6.2.1 The Symmetric Object

In this section we parameterise three views of the symmetric object. The three sample views are at $Y_v = 0$, $Y_v = 1$ and $Y_v = 2$. The solution places the three views at parameter values $u_1 = -0.7076$, $u_2 = 0.0009$ and $u_3 = 0.7067$. If we scale and shift the u_i values so that $u_1 = 0$ and $u_2 = 1$ correspond to the Y_v positions of the first two views, then we find that $u_3 = 1.9962$, which is close to the actual value of $Y_v = 2$ for the third target view.

We will now use the solution for each of the α_j, β_j and γ_j to compute the singular values w_4, w_5 and w_6 and the value of $l_1 m_2 - l_2 m_1$ over the range $u = -0.7076$ to $u = 2.8347$. This range of u values corresponds to the range $Y_v = 0$ to $Y_v = 5$.

Figure 6.12 shows the three smallest singular values over the range $u = -0.7076$ to $u = 2.8347$ ($Y_v = 0$ to $Y_v = 5$). To allow an easy comparison with figure 6.2 the scale is in terms of Y_v. We can see from figure 6.12 that the two points at which the singular values meet are at $Y_v \approx 0.825$ and $Y_v \approx 3.575$, which are close to the actual values in figure 6.3 of $Y_v \approx 0.85$ and $Y_v \approx 3.7$.

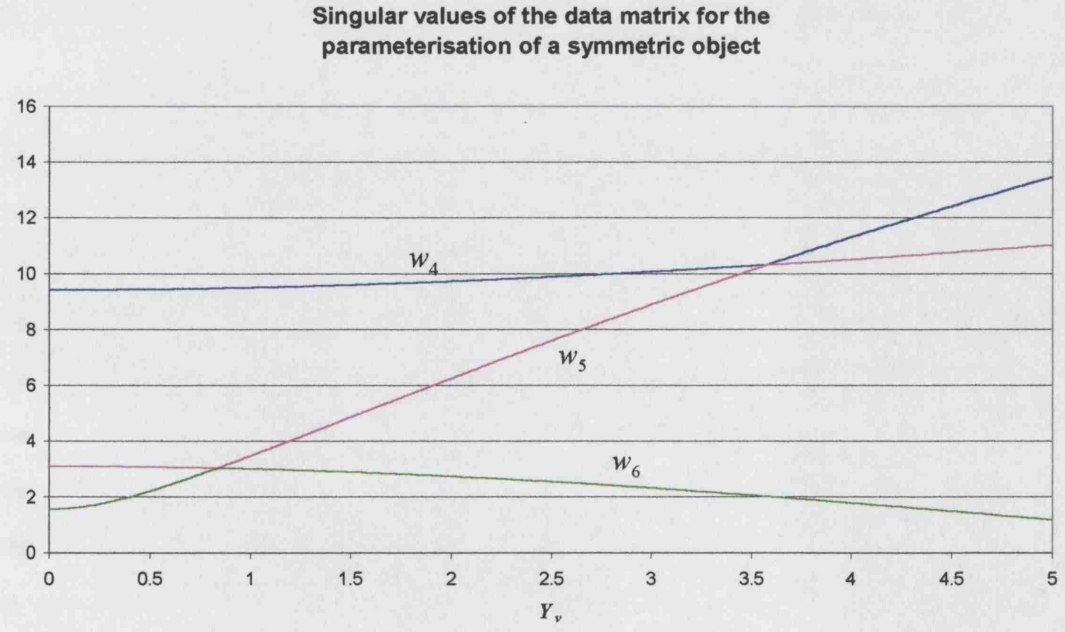


Figure 6.12. The singular values for the parameterisation of the symmetrical object.

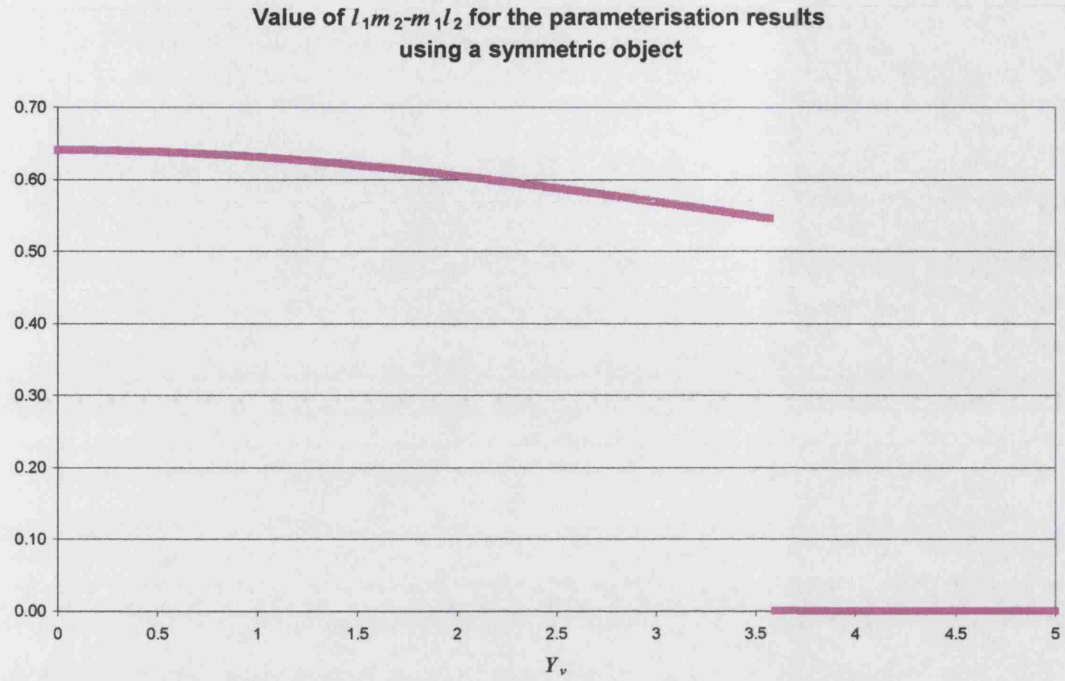


Figure 6.13. $l_1m_2 - m_1l_2$ for the parameterisation of the symmetrical object

Figure 6.13 shows the absolute value of $l_1m_2 - l_2m_1$ over the range $u = -0.7076$ to $u = 2.8347$ ($Y_v = 0$ to $Y_v = 5$). As in figure 6.5 the value of $l_1m_2 - l_2m_1$ drops to zero and stays there. From figures 6.12 and 6.13 we can see that the critical point is at $Y_v \approx 3.575$, which is close to the actual critical point in figures 6.3 and 6.5 at $Y_v \approx 3.7$.

6.2.2 The Asymmetric Objects

In this section will repeat the experiment in section 6.2.1 for the nearly symmetrical and random objects described in section 6.1.2. The sample views in each case were taken, as in the previous example, at distances of $Y_v = 0$, $Y_v = 1$ and $Y_v = 2$.

In the case of the nearly symmetrical object the parameterisation places the sample views at parameter values of $u_1 = -0.7075$, $u_2 = 0.0007$ and $u_3 = 0.7067$. If we scale and shift the u_i values such that $u_1 = 0$ and $u_2 = 1$, so that they correspond to the Y_v values of the first two sample views, we find that the value of u_3 is 1.9968. This value of $u_3 = 1.9968$ is close to the actual value of $Y_v = 2$.

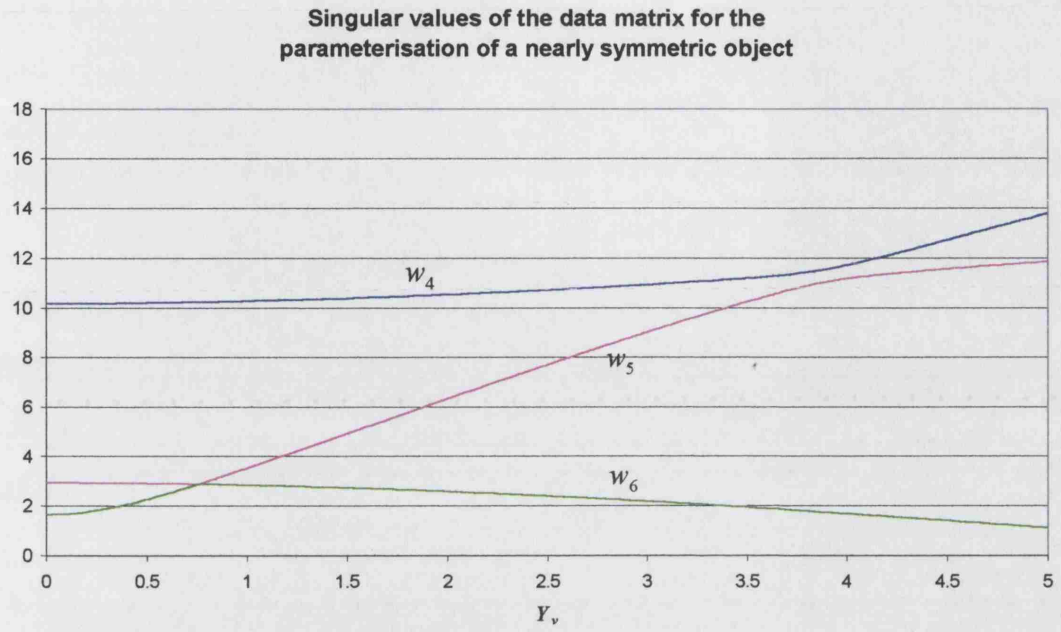


Figure 6.14. Singular values for the parameterisation of the nearly symmetrical object.

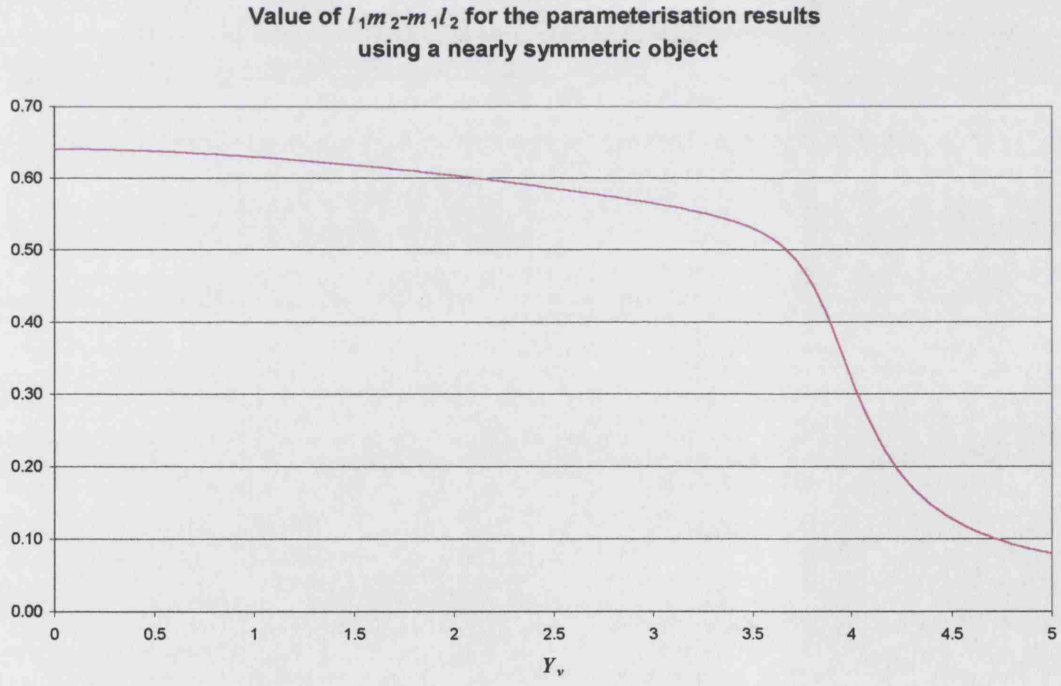


Figure 6.15. $l_1m_2 - m_1l_2$ for the parameterisation of the nearly symmetrical object.

We will now use the result of the parameterisation to predict the singular values and the value of $l_1m_2 - l_2m_1$ for the near symmetric object over the range

$u = -0.7075$ to $u = 2.8335$ which corresponds to a range of $Y_v = 0$ to $Y_v = 5$. Figure 6.14 shows the singular values and figure 6.15 shows the absolute value of $l_1 m_2 - l_2 m_1$. It can be seen that the curves in figures 6.14 and 6.15 look similar to the curves in figures 6.6 and 6.7. The parameterisation has correctly predicted the relative positions of the sample views and the shape of the curves.

The parameterisation of the views of the random object places the three sample view at parameter values $u_1 = -0.7034$, $u_2 = -0.0073$ and $u_3 = 0.7107$. If we scale the values of u so that $u_1 = 0$ and $u_2 = 1$ so that they correspond to the Y_v values of the first two sample views, we find that $u_3 = 2.0315$ which is close to the actual value of $Y_v = 2$.

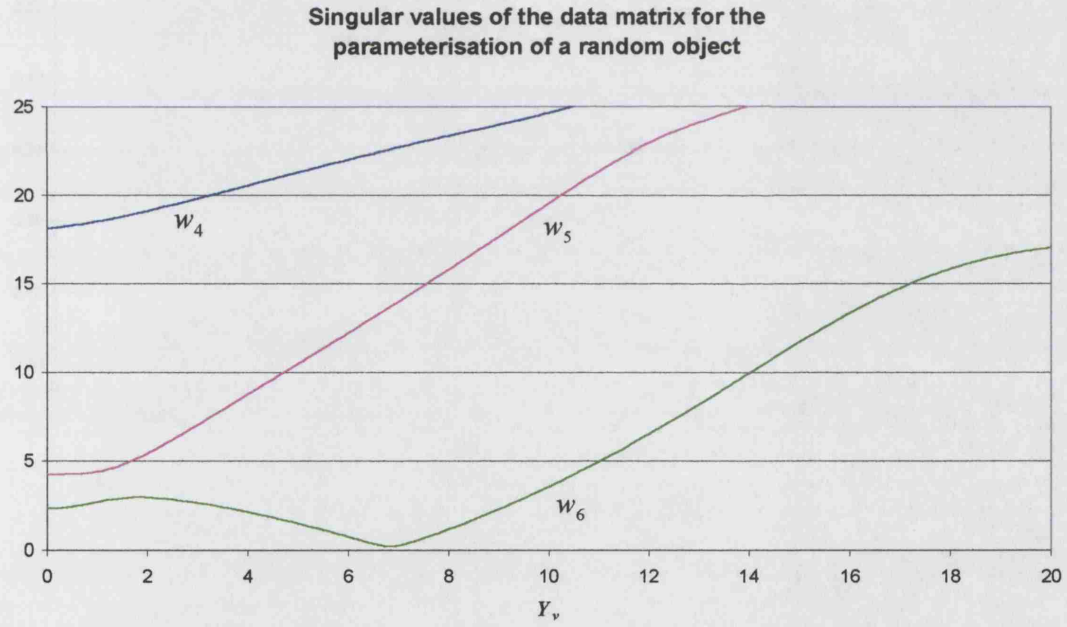


Figure 6.16. Singular values for the parameterisation of the random object.

We will now use the result of the parameterisation to predict the singular values and the value of $l_1 m_2 - l_2 m_1$ for the random object over the range $u = -0.7034$ to $u = 13.21891$ which corresponds to a range of $Y_v = 0$ to $Y_v = 20$. Figure 6.16 shows the singular values and figure 6.17 shows the absolute value of $l_1 m_2 - l_2 m_1$. It can be seen that the curves in figures 6.16 and 6.17 look similar to the curves in

figures 6.10 and 6.11 over the range $Y_v < 5$. Over the range $5 < Y_v < 20$ we can see the curves in figures 6.17 and 6.16 no longer look similar to those in figures 6.10 and 6.11. To show how the errors in the curves increase, figure 6.18 shows the values of $\Delta w_i = w_i^p - w_i^r$ for $i = 4..6$, where w_i^p is the i^{th} singular value of the result of the parameterisation (as in figure 6.17) and w_i^r is the actual i^{th} singular value (as in figure 6.10).

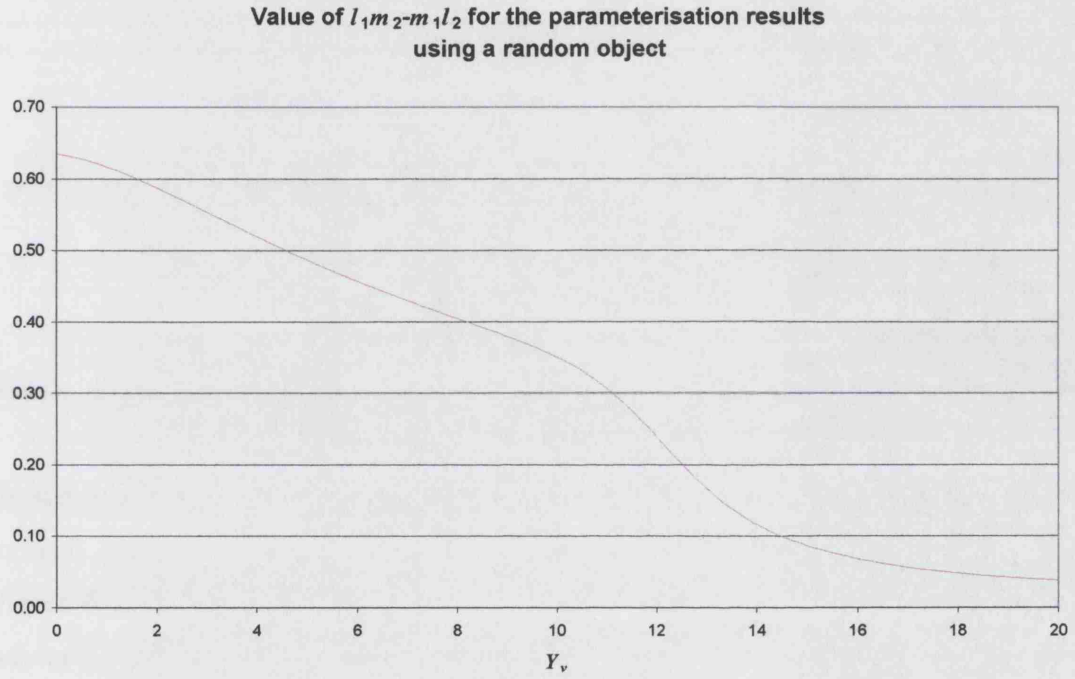


Figure 6.17. $l_1m_2 - m_1l_2$ for the parameterisation of the random object.

From figure 6.18 we can see that the parameterisation works well in the range $Y_v < 5$. After $Y_v = 5$ the errors in the predicted singular values increase rapidly. Therefore with the sample views at positions of $Y_v = 0$, $Y_v = 1$ and $Y_v = 2$ the parameterisation works well until $Y_v = 5$.

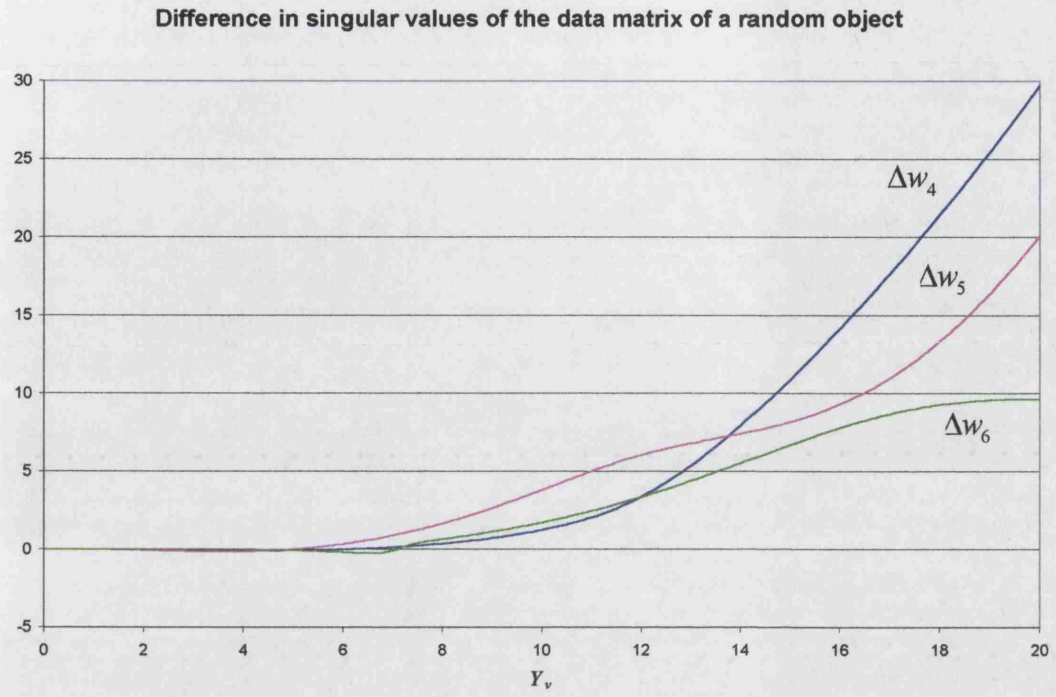


Figure 6.18. Difference between the predicted and actual singular values.

6.2.3 Real Images

In this section we use a set of real symmetric sample images to determine whether a critical viewpoint exists. The sample images of the symmetric test object are shown in figure 6.19. The basis views are shown in figure 6.19 (a) and (b) and the three target views are shown in (c), (d) and (e).

The images were taken with the camera in a fixed position. The test object was rotated left and right for the two basis views and then tilted backwards to three different positions for the target views. The control points were chosen to be the corners of each of the eight planes and the point at the tip of the tiger's tooth in each of the planes. The control points were located by hand. The three target images were then parameterised using the positions of the control points in each of the views. The parameterisation placed the three sample target views in figure 6.19 (c), (d) and (e) at parameter values $u = -0.6922$, $u = -0.0290$ and $u = 0.7212$ respectively. The results of the parameterisation and the control points in the basis views were then used to determine the singular values and the value of $l_1 m_2 - l_2 m_1$ for target views in the range $-3 < u < 3$.

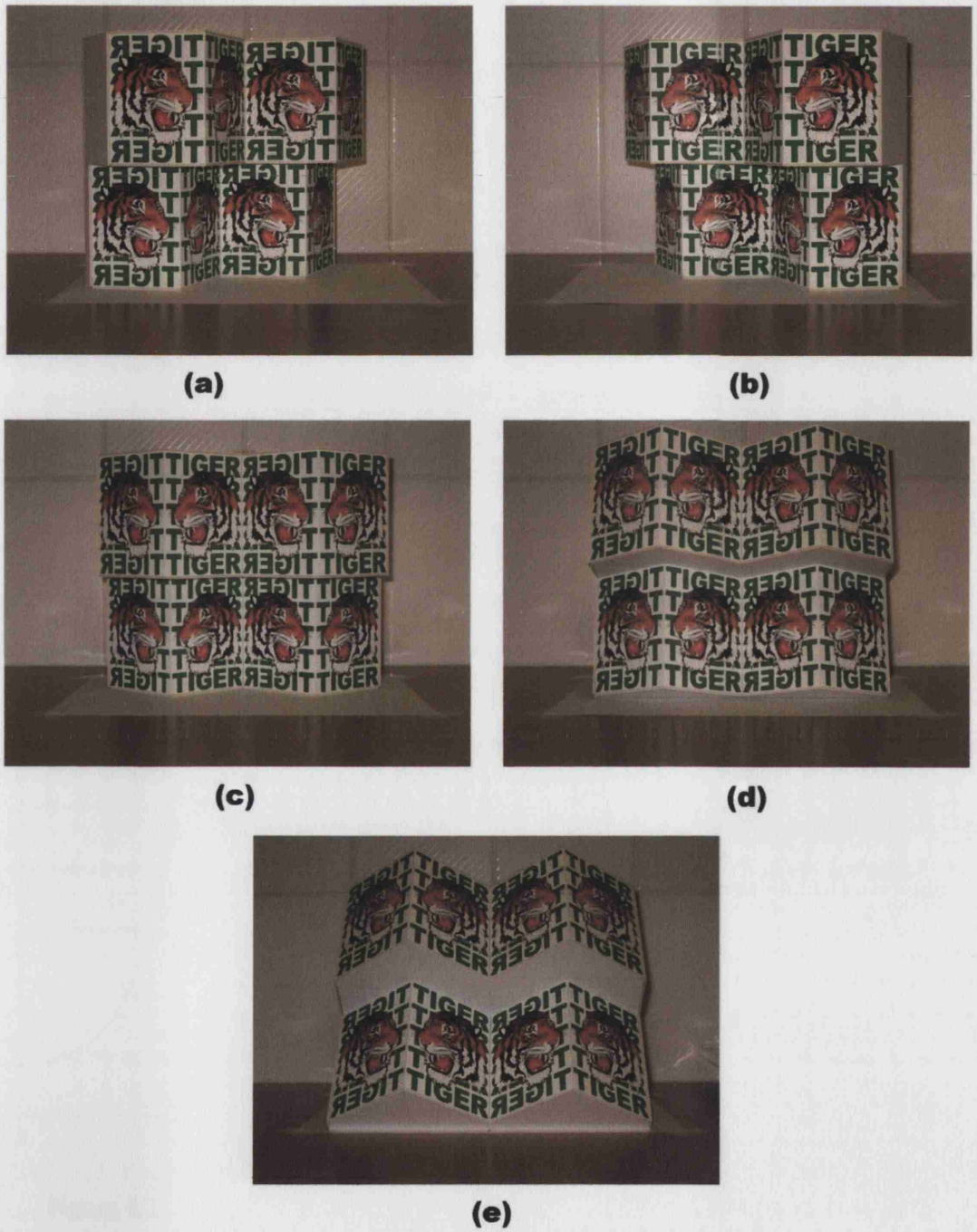


Figure 6.19. Sample images of a symmetrical test object.

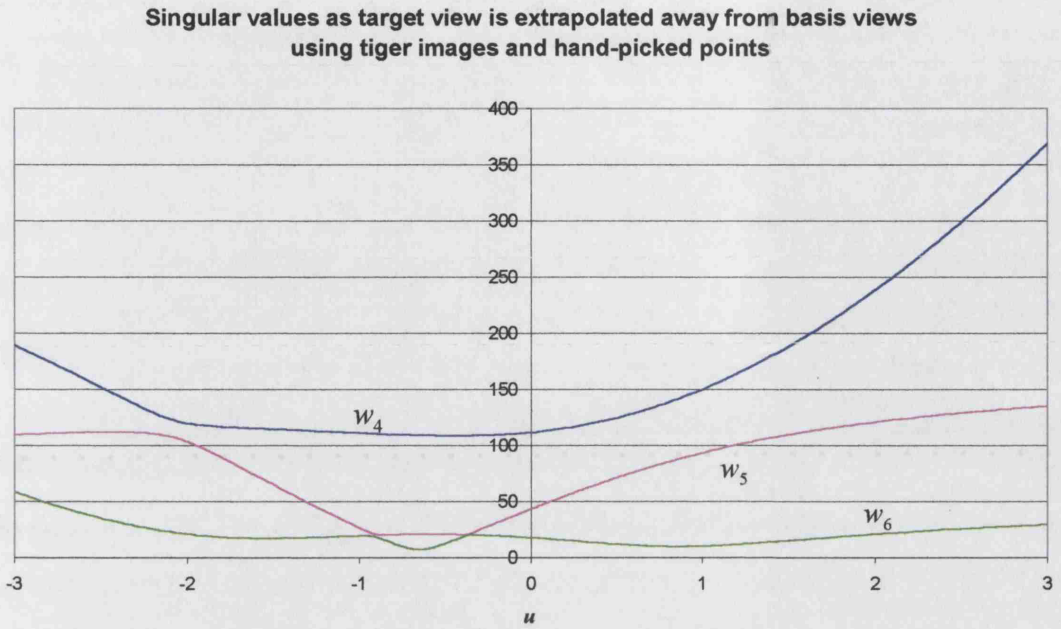


Figure 6.20. Singular values for the tiger images using hand-picked control points.

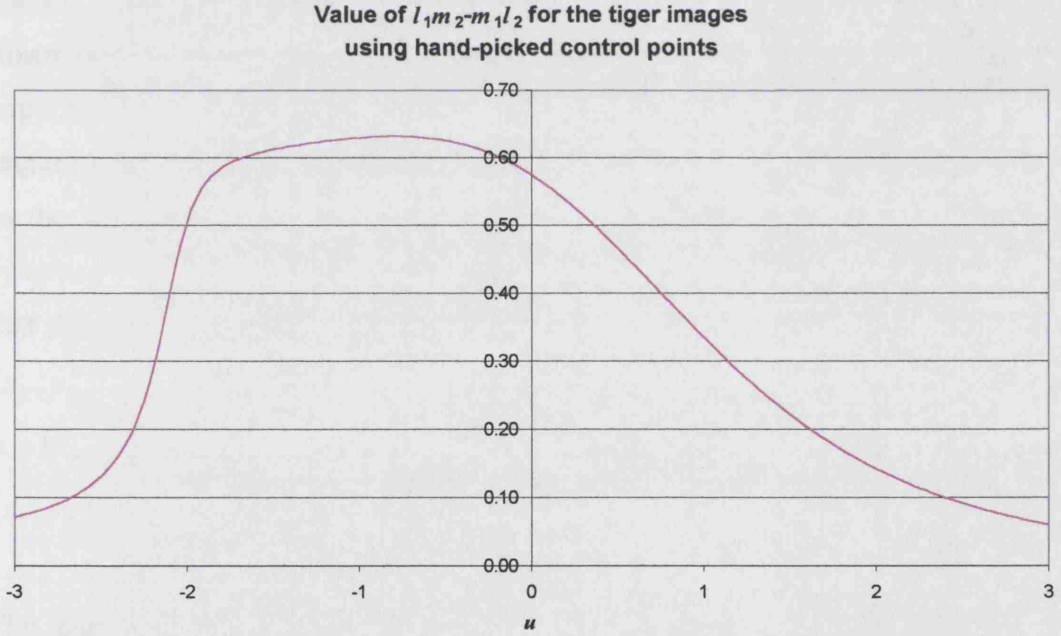


Figure 6.21. $l_1m_2 - m_1l_2$ for the tiger images using hand-picked control points.

Figure 6.20 show the singular values and figure 6.21 shows the absolute value of $l_1m_2 - l_2m_1$ for target views in the range $-3 < u < 3$. When we extrapolate away from the midpoint, $u = -0.69$, towards $u = -3$ we can see that singular values w_5 and w_6 appear to coincide at $u = -0.90$. In fact they do not, they get close and then each

of the curves changes direction. We can also see that at $u = -2.07$ the singular values w_4 and w_5 get close and then change direction. This is similar to what we have seen previously when we extrapolate away from the baseline for the near symmetrical object in figure 6.6. This would indicate that the images are of a near symmetrical object and we can use point $u = -2.07$ as a limit of extrapolation. This limit of extrapolation can also be seen in figure 6.21 where the value of $l_1 m_2 - l_2 m_1$ drops rapidly at $u = -2.07$.

When we extrapolate in the opposite direction (from $u = -0.69$ to $u = 3$) we can see that the singular values w_5 and w_6 appear to touch at $u = -0.36$. Again they do not actually meet they get close and change direction. The singular values w_4 and w_5 do not get close in the range $-0.36 < u < 3$. In this direction it is not possible to determine an obvious limit of extrapolation from the singular values.

As we extrapolate from $u = -0.69$ to $u = -3$ the singular values indicate that we have a near symmetrical object. This may be due to errors on the locations of the control points which means that the symmetry relationships given in 6.1.1 and illustrated in figure 6.2 will not be satisfied exactly. We will now repeat the experiment using control points that have been adjusted so that they do satisfy the required symmetry relationships. To do this we use a mirror image of figure 6.19 (a) as the second basis view and calculate the control points in this view by using the symmetry relationships. The control points in each of the target views are adjusted so that they satisfy the symmetry relationships. To do this we replace a pair of points (x_R, y_R) and (x_L, y_L) (where $x_L \approx -x_R$ and $y_R \approx y_L$) with (x, y) and $(-x, y)$, where $x = (x_R - x_L)/2$ and $y = (y_R + y_L)/2$. Points that lie on the vertical line of symmetry, (x_S, y_S), are replaced with $(0, y_S)$.

We then parameterised the three target views using the adjusted control points. The parameterisation placed the three views at parameter values $u = -0.6922$, $u = -0.0288$ and $u = 0.7211$ which are similar to the values above which were obtained by using the hand-picked points. We then plot curves of the three smallest singular values and the absolute value of $l_1 m_2 - l_2 m_1$ for values of u in the range -3 to 3. Figure 6.22 shows the three smallest singular values and figure 6.23 shows the absolute value of $l_1 m_2 - l_2 m_1$.

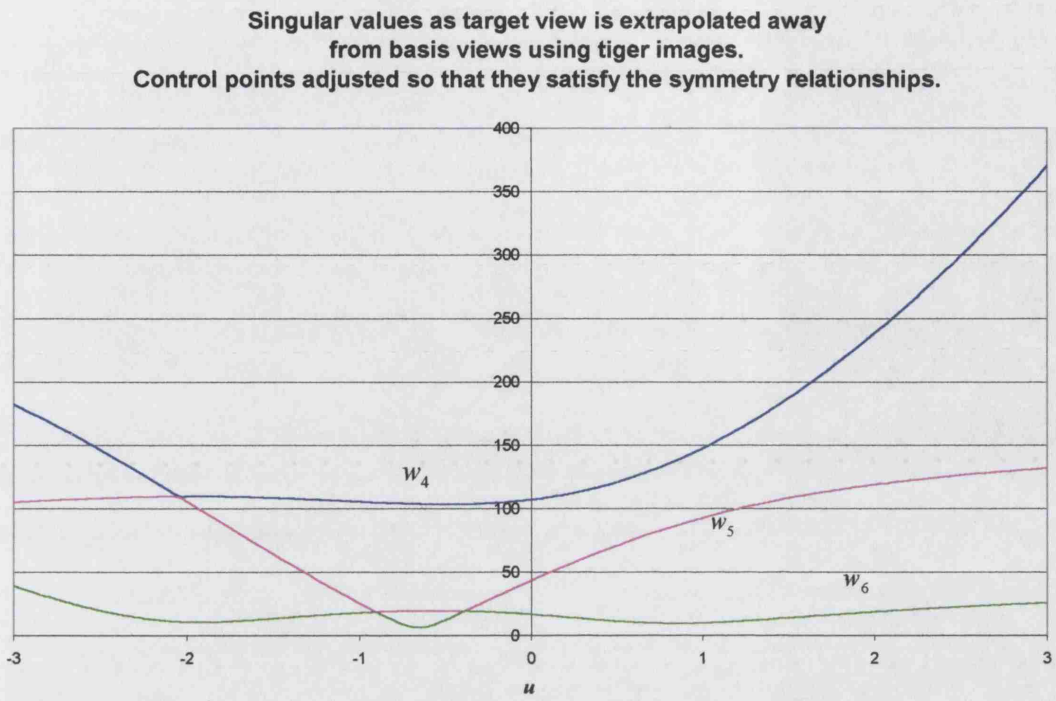


Figure 6.22. Singular values for the tiger images using adjusted control points.

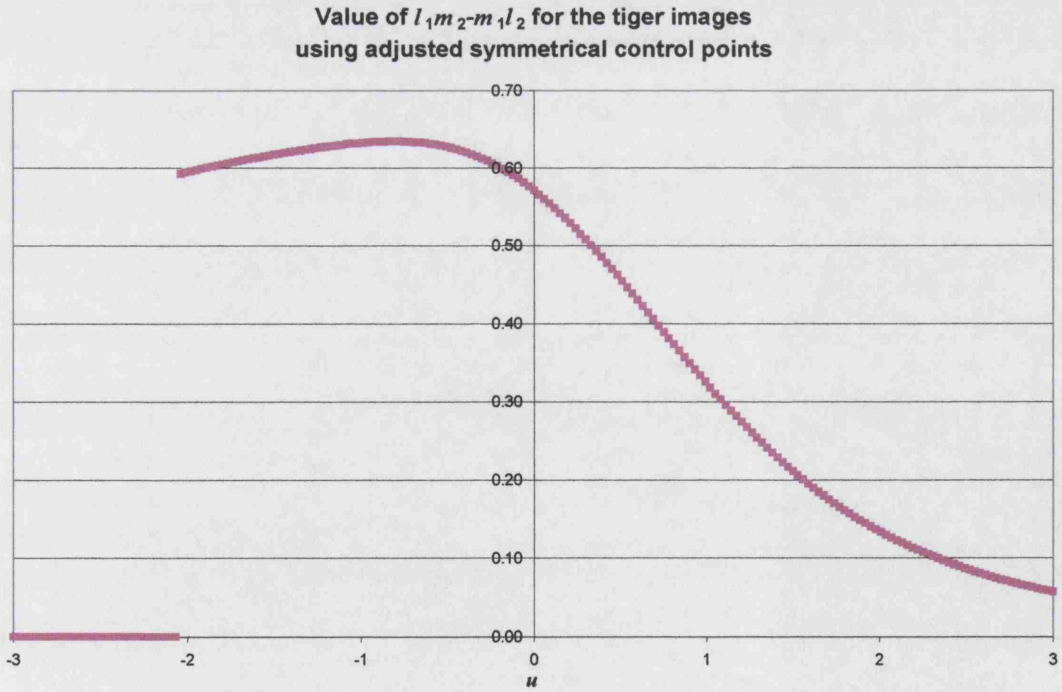


Figure 6.23. $l_1m_2 - m_1l_2$ for the tiger images using adjusted control points.

We can see, from figure 6.22, that curves of the singular values w_5 and w_6 touch at $u = -0.93$ and $u = -0.39$, and the curves of the singular values w_4 and w_5 touch at a distance of $u = -2.07$. It can also be seen, from figure 6.23, that the value of $l_1 m_2 - l_2 m_1$ drops to zero at $u = -2.07$. This indicates that there is a critical viewpoint at $u = -2.07$. In this case there is not a critical viewpoint when we extrapolate from $u = -0.6922$ to $u = 3$. We mentioned earlier that the symmetry relationships allow a critical viewpoint to exist, but this does not mean that there has to be a critical viewpoint in every symmetric case. In this example we can see a critical viewpoint exists when we extrapolate in one direction, but not when we extrapolate in the opposite direction.

If we compare figures 6.22 and 6.23 with 6.20 and 6.21 respectively we can see that in the range $-1.89 < u < 3$ the curves look similar. It is only around the critical viewpoint that the curves look different.

We complete this section by generating some novel views in the range $-2 < u < 1.5$. These novel views are shown in figure 6.24. The sample views in figure 6.19 (a) and (b) are used as the basis views. The control points in the novel views are calculated using the result of the parameterisation of the hand-picked control points of the three sample views (figure 6.19 (c), (d) and (e). It can be seen that all of the views in figure 6.24 look realistic. The novel view at $u = -2$ appears slightly less realistic than the other views as the lines at the bottom of each of the faces of the object do not appear parallel in this image whereas they are for each of the sample views. It is likely that this deterioration in quality is caused by the fact that we have extrapolated further away from the original sample views (at u values of -0.6922, -0.0290 and 0.7212) in the direction of $-u$ than in the direction of $+u$, rather than the existence of a critical viewpoint. When we extrapolate in the opposite direction (towards $u = 1.5$) the view synthesis procedure breaks down because each of the faces become inverted beyond approximately $u = 2$ so that we would be imaging the back of each of the faces of the object. Although the positions of the control points can still be calculated for $u > 2$ the faces of the test object can no longer be rendered.

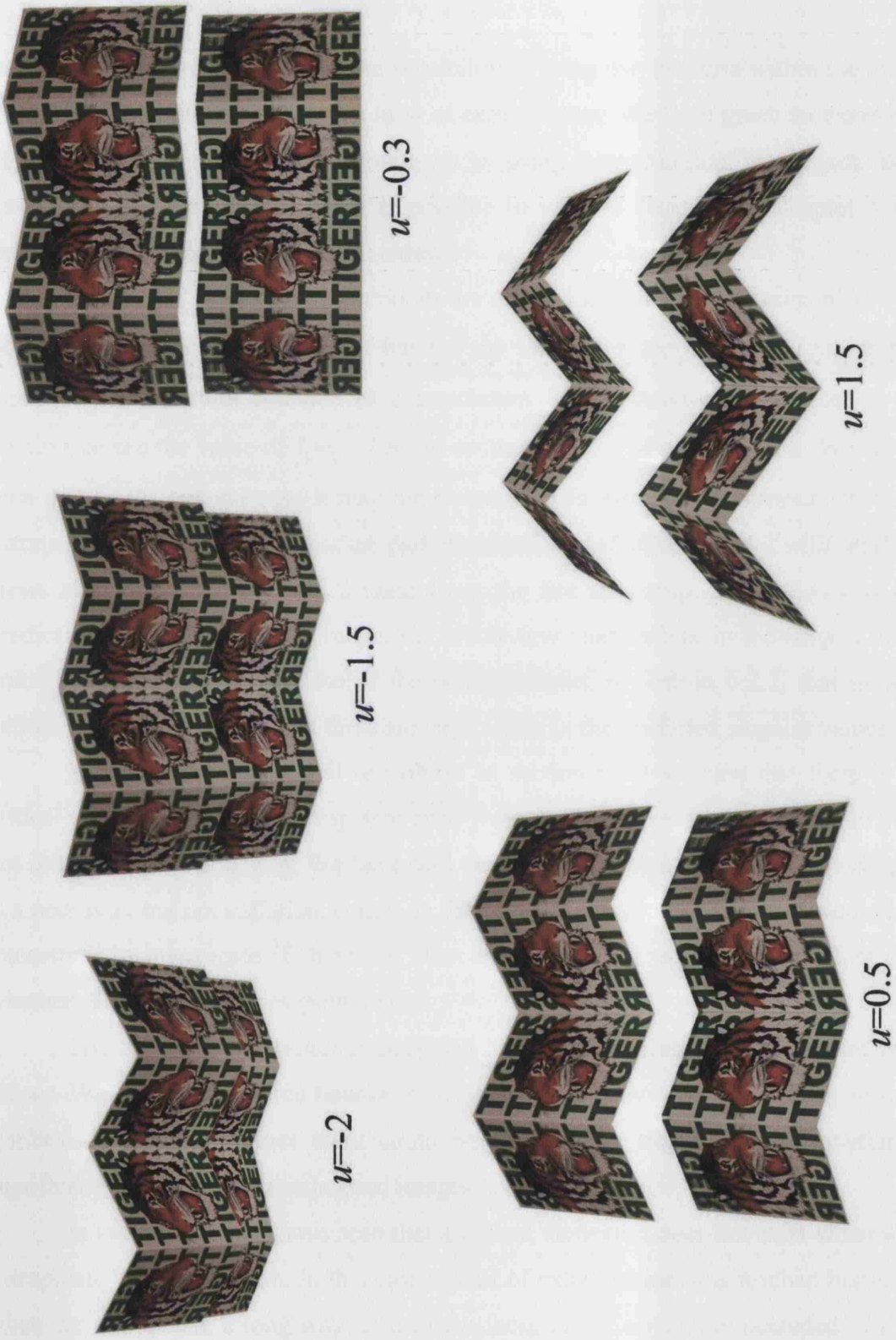


Figure 6.24. Novel views of the symmetrical test object.

6.3 Conclusions and Further Work

In this chapter we have explored the possibility of using the structure within the total least squares solution to estimate a limit of extrapolation. We have given an example of when the TLS relationships breakdown by using a symmetrical test object. We have shown in section 6.2 that it is possible to use the theory from chapter 5 to determine the critical viewpoint if it exists.

In the case where the object points are close to a symmetrical arrangement we no longer see a critical viewpoint but we see that the value of $l_1 m_2 - l_2 m_1$ drops rapidly after a certain distance of extrapolation. Under these circumstances it is possible to use the value of $l_1 m_2 - l_2 m_1$ to estimate a limit of extrapolation. We have seen that in the general case it may not be possible to estimate an obvious limit of extrapolation. We have seen that the parameterisation used in section 6.2 with sample views at distances of 0, 1 and 2 units along the line of extrapolation allows us to predict accurately the singular values and multi-view relationships in the range 0 to 5 units. We have seen in the case of the random object, in section 6.2.2, that as we extrapolate beyond 5 units that there are large errors in the predicted singular values.

In the example of a real test object in section 6.2.3 we saw that there is a critical viewpoint when we extrapolate away from the basis views in one direction but not in the opposite direction. We have said that satisfying the symmetry relationships is a necessary but not sufficient condition for a critical viewpoint to exist. It would be interesting to investigate if there are other conditions that may have an effect on whether or not a critical viewpoint exists.

The limits of extrapolation discussed here are based on a breakdown of the theory. We have not explored how or if this relates to a deterioration in quality of the synthesised images. Further work could be done to investigate whether anything significant happens in the synthesised images at the critical viewpoint.

In section 6.2.3 we have seen that a critical viewpoint does not exist when we extrapolate in one direction. In this case a limit of extrapolation was reached because when we extrapolate a long way each of the faces of the object are occluded in the target image. It may be possible therefore to determine a limit of extrapolation by plotting the positions of the control points in the target views and looking at where

any occlusions will occur. It is worth investigating this possibility as it would not be limited to symmetrical objects and could be used in the general case.

It may also be possible to determine a limit of extrapolation by looking at how the quality of the synthesised images deteriorates. It would be interesting to investigate how far we can extrapolate in the general case before the quality of the synthesised images becomes unacceptable.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis we have given a method of synthesising novel views using only a set of sample images and information that can be obtained from those images. The sample images are obtained using uncalibrated cameras and without us having any information about the positions of the cameras or any measurements of the objects in the images.

In order to develop the view synthesis method we have made some assumptions about the images. The first assumption is that all the images are obtained under affine imaging conditions. It is advantageous to make this assumption as the mathematics needed to form the multi-view relationships is much simpler than in the perspective case. The affine multi-view relationships provide a good approximation to the perspective relationships and still allow us to synthesise realistic views.

The next assumption we make is that the objects are locally planar. This assumption is used when rendering the intensities in a synthesised target view. The images are divided into triangular planar facets. The intensities in each of the triangles in the target view are rendered using a combination of the intensities from the basis views.

In chapter 5 we have described a method of synthesising novel views starting from a set of sample views. To develop the method we begin by parameterising the sample views in terms of two parameters. This assumes that the images are obtained using an affine camera and that all the images are obtained with the camera in the same orientation, i.e., there is no rotation about the Z axis of the camera.

7.1.1 Contributions

In chapter 1 we gave an introduction to the thesis and chapters 2 and 3 provided background information about the geometry of multiple views. The main contributions of this thesis are contained within chapters 4, 5 and 6. Here we provide a list of contributions for each of these three chapters.

The first main contribution that we make is the method that we use to estimate the affine multi-view relationships. In chapter 4 we form new affine multi-view relationships and show how they can be used to encode target views as a combination of a pair of basis views. We will now list the contributions that are made in chapter 4.

1 We form a new pair of affine multi-view relationships that treat each of the co-ordinates in each of the views (the target view and each of the basis views) in the same way. This pair of affine multi-view relationships has the advantage that they are symmetric in each of the views. They are also less sensitive to any errors in the positions of the control points than the asymmetric relationships.

2 We solve for the co-efficients of the multi-view relationships using a total least squares procedure instead of an ordinary least squares procedure. The total least squares method allows us to treat each of the control points in each of the views (including the target view) in a similar fashion. This is the correct way to solve for the relationships when the target view is one of the existing views and all the control points in the target view and each of the basis views are located using the same method. If this is the case each of the views are equally likely to contain measurement errors.

3 We give a method of rendering the intensities in the target view as an interpolation of the basis view intensities using the co-efficients of the total least squares relationships. The method allows us to produce realistic images and is consistent with the methods previously used with the ordinary least squares relationships.

The next contribution that we make is the method of synthesising a novel view that is developed in chapter 5. This can be broken down into stages as follows:

4 We parameterise a set of sample view by finding a mapping between a set of variables, E_j, to a pair of parameters (u and v). This parameterisation can be used to parameterise any of the variables of the sample views that vary smoothly as the target view is moved throughout the viewspace. In chapter 5 we have shown how the method can be used to parameterise three different sets of variables. The three choices for the E_j are:

I. The elements of the matrix R obtained from the Cholesky decomposition of the data matrix $D^T D$ formed when finding the total least squares relationships.

II. The co-efficients of the least squares relationships.

III. The co-ordinates of the control points.

5 The parameterisation provides us with the relative positions of the sample views in the 2D parameter plane. We can choose the positions of the novel views relative to the sample views in the parameter plane and read off the values of the parameters u and v. These values can be used to calculate the sets of view variables for each of the novel views.

6 We can use the values of the view variables to synthesise novel views at various positions in the parameter plane. The method allows us to extrapolate away from the sample views as well as interpolate between them.

7 We show that when we parameterise the sample images by using the positions of the control points we can synthesise novel views that are more accurate and more realistic in appearance than when we parameterise the co-efficients of the multi-view relationships or the Cholesky decomposition of the data matrix $D^T D$.

8 In chapter 6 we show how the parameterisation can be adapted for only one parameter, for example, when the camera is moving along a line.

The final contribution that we make is in chapter 6 where we explore the possibility of using the structure within the total least squares problem to determine a limit of extrapolation in the case of a symmetric object.

9 We have shown that in the special case where there is a high level of symmetry in the configuration of the cameras and in the positions of the control points in space a critical viewpoint may exist where the total least squares relationships breakdown.

10 We have shown that the critical viewpoint exists for a set of real symmetric images and that it is possible to estimate its position using the parameterisation described in chapter 5.

7.2 Further Work

The starting point for the view synthesis methods in this thesis is a set of sample images and a set of control points across the images. Throughout this thesis we have located the control points by hand. It would be useful if we could locate a set of matching control points in the sample views automatically.

In chapter 4 we have used the total least squares procedure to solve for the affine multi-view relationships. We also introduced the generalised total least squares procedure and said that this should be used when the errors are not independent and not identically distributed across all the points. The generalised total least squares method could be used to determine the perspective multi-view relationships by choosing an appropriate weighting of the errors. The generalised total least squares procedure should also be used when we are synthesising novel views by using an interpolation scheme (for example the scheme described in chapter 5) to locate the positions of the control points in the novel view. In this case the control points in the novel view will still contain errors but these errors will be a function of the errors on the sample views that we are interpolating between.

In chapter 5 we have described a method of parameterising a set of sample views in terms of the control points. This assumes that all the control points are visible in each of the sample views. This may not be the case if the images used have

a high number of occlusions. We should consider what happens when there are occlusions and points may be missing from one or more of the views.

Throughout chapter 5 we have parameterised the sample views in terms of two parameters. By using two parameters we have assumed that the images have been obtained using affine camera in the same orientation (section 5.2). We know that the general perspective camera has eleven degrees of freedom. It would be interesting and useful to investigate how this parameterisation scheme could be extended to include a higher number of parameters.

In chapter 6 we have discussed the possibility of using the structure within the total least squares solution to determine a limit of extrapolation. We have not explored whether the breakdown of the total least squares relationships relates to any deterioration in the quality of the synthesised images.

It would also be interesting to investigate other methods for estimating a limit of extrapolation. One possibility for doing this is by exploring any occlusions that arise in the image. We mentioned in section 6.2.3 that the view synthesis procedure breaks-down in one direction because each of the faces of the object become inverted when we extrapolate beyond a certain point. Although the parameterisation allows us to locate the positions of the control points in views beyond this point the views cannot be rendered because none of the faces can be seen in these views.

Appendix A

Singular Value Decomposition and Solving Least Squares Problems

The singular value decomposition (SVD) of a $n \times m$ matrix, D is written as the product of three matrices:

$$D = UWV^T \qquad \text{(A.1)}$$

where U is a $n \times m$ orthogonal matrix, V^T is a $m \times m$ orthogonal matrix and W is a $m \times m$ diagonal matrix, $W = \text{diag}(w_1 \quad w_2 \quad . \quad . \quad w_m)$ [VV91, PTVF93]. The entries of W are either positive or zero and correspond to the singular values of the matrix D. It is usual to assume that the matrix W is arranged such that $w_1 \geq w_2 \geq ... \geq w_m$.

The singular values of the matrix D are the non-negative square roots of the eigenvalues of the matrix $D^T D$ [Lüt96]. We can write V as $(\underline{v}_1 \quad \underline{v}_2 \quad . \quad . \quad \underline{v}_m)$, where each column $\underline{v}_i$ is the singular vector [GV96] of the matrix D that corresponds to the singular value w_i. The singular vectors of the matrix D are also the eigenvectors of the matrix $D^T D$.

A.1 Solution of Least Squares Problems

Given a problem of the form:

$$\underline{x} + \underline{\varepsilon} = D\underline{a} \quad , \qquad \text{(A.2)}$$

where the matrix D and the vector $\underline{x}$ are known. We want to find a solution for the parameter vector $\underline{a}$ such that the total sum of squared errors, $\|\underline{\varepsilon}\|^2$ is minimised. We can re-arrange (A.2) to get an expression for $\underline{\varepsilon}$.

$$\underline{\varepsilon} = D\underline{a} - \underline{x} \quad . \tag{A.3}$$

If we square both sides of (A.3), differentiate with respect to $\underline{a}$ and set the derivative equal to zero, we obtain the normal equation:

$$D^T D\underline{a} = D^T \underline{x} \quad . \tag{A.4}$$

By rearranging (A.4) we can obtain an expression for $\underline{a}$:

$$\underline{a} = \left(D^T D\right)^{-1} D^T \underline{x} \quad . \tag{A.5}$$

By substituting the SVD of the matrix D into equation (A.5) we can express the solution for $\underline{a}$ terms of U, W and V^T.

$$\underline{a} = \left(VWU^T UWV^T\right)^{-1} VWU^T \underline{x} \tag{A.6}$$

Since U and V^T are orthogonal, we know that $UU^T = U^T U = VV^T = V^T V = I$, where I is the identity matrix, and that $V^{-1} = V^T$. We also know from the properties of inverses that $\left(VWWV^T\right)^{-1} = VW^{-1}W^{-1}V^T$ [Lüt96]. By using these properties we can simplify (A.5):

$$\underline{a} = VW^{-1}U^T \underline{x} \quad . \tag{A.7}$$

If the matrix D is non-singular then each of the singular values are non-zero and we can compute the inverse of W as $\mathrm{diag}(1/w_1 \quad 1/w_2 \quad . \quad . \quad 1/w_5)$ [Lüt96]. In the case where D is singular (or close to singular) the solution is ill conditioned and one or more of the singular values will be equal to zero (or close to zero). In this case it is possible to find a solutions for $\underline{a}$ by computing the pseudo inverse of the matrix D. This corresponds to replacing W^{-1} with $\mathrm{diag}\left(w_1^+ \quad w_2^+ \quad . \quad . \quad w_5^+\right)$ in (A.7):

$$\begin{aligned} \underline{a} &= V\mathrm{diag}\left(w_1^+ \quad w_2^+ \quad . \quad . \quad w_5^+\right)U^T \underline{x} \\ \underline{b} &= V\mathrm{diag}\left(w_1^+ \quad w_2^+ \quad . \quad . \quad w_5^+\right)U^T \underline{y} \end{aligned} \quad , \tag{A.8}$$

where $w_i^+ = \begin{cases} 1/w_i & w_i > 0 \\ 0 & w_i = 0 \end{cases}$.

A.2 Solution of Homogeneous Linear System of Equations

In the case of a homogeneous system of equations the least squares solution can be found by using singular value decomposition of the data matrix. For example consider the linear system of equations in (3.4.10) for the fundamental matrix:

$$A\underline{f} = \underline{\varepsilon} \quad . \tag{A.9}$$

We want to find a solution for $\underline{f}$ such that $\|\underline{\varepsilon}\|^2$ is minimised. A solution can be found by performing a SVD of the data matrix A.

$$A = UWV^T \tag{A.10}$$

The solution for the vector $\underline{f}$ is equal to the singular vector of A corresponding to the smallest singular value [LH95].

Appendix B

The Mixed Least Squares-Total Least Squares Problem

The mixed least squares-total least squares (mixed LS-TLS) problem arises when we have a system of linear equations where one or more columns of the data matrix are known exactly [VV91]. For example, suppose we have a set of n equations in $m \times d$ unknowns X:

$$AX \approx B \quad , \tag{B.1}$$

where A is an $n \times m$ data matrix those first m_1 columns are error free and B is a $n \times d$ matrix that also contains measurement errors. We wish to find the best estimate for X, such that the error on the relationship (B.1) is minimised. If the errors in A_2 and B are uncorrelated and of equal size then (B.1) is a mixed least squares-total least squares (mixed LS-TLS) problem [VV91].

The first stage of the mixed LS-TLS solution is to partition the matrices. We partition A as $(A_1 \quad A_2)$, where A_1 is a $n \times m_1$ matrix whose entries are error free and A_2 is a $n \times m_2$ matrix that contains measurement errors. Similarly we partition X as $(X_1 \quad X_2)$, where X_1 is a $m_1 \times d$ matrix and X_2 is $m_2 \times d$. We then perform m_1 (the number of error free columns in A) householder transformations [GV99, LH95], Q on the matrix $(A_1 \quad A_2 \quad B)$ such that:

$$Q^T(A_1 \quad A_2 \quad B) = \begin{pmatrix} R_{11} & R_{12} & R_{1b} \\ 0 & R_{22} & R_{2b} \end{pmatrix}. \tag{B.2}$$

Where R_{11} is a $m_1 \times m_1$ upper triangular matrix, R_{12} is $m_1 \times m_2$, R_{22} is $n - m_1 \times m_2$, R_{1b} is $m_1 \times d$ and R_{2b} is $n - m_1 \times d$. Since matrix Q is an orthogonal matrix, we can multiply $(A_1 \quad A_2 \quad B)$ by the matrix Q^T without affecting the distribution of the errors.

The solution for X_2 can then be found by solving:

$$R_{22}X_2 = R_{2b} \tag{B.3}$$

as a total least squares problem [VV91]. Once we have found the solution for X_2, the solution for X_1 can be found by solving the equation:

$$R_{11}X_1 + R_{12}X_2 = R_{1b} \quad . \tag{B.4}$$

Together the matrices X_1 and X_2 provide a mixed LS-TLS solution to the problem (B.1).

B.1 The Mixed LS-TLS Problem and the Affine Multi-view Relationships

In chapter 4, section 4.2.1 we introduced a pair of affine multi-view relationships:

$$\begin{aligned} l_1x_i + l_2y_i + l_3x_i' + l_4y_i' + l_5x_i'' + l_6y_i'' + l_7 &= \varepsilon_i \\ m_1x_i + m_2y_i + m_3x_i' + m_4y_i' + m_5x_i'' + m_6y_i'' + m_7 &= \eta_i \end{aligned} \quad . \tag{B.5}$$

We solved for the coefficients l_i and m_i by summing over all the control points to find expressions for l_7 and m_7, substituting these expressions back into the equations (B.5) and then solving as a total least squares problem. We will now shown that if equations (B.5) are solved using the mixed LS-TLS method described above, the resulting expressions for l_7 and m_7 are the same as those obtained in section 4.2.1 (equations (4.2.8)) and hence that the procedure used in section 4.2.1 leads to a valid solution for the co-efficients.

If we re-arrange (B.5) into matrix form as:

$$\begin{aligned} D\underline{l} &= (D_1 \quad D_2)(\underline{l}_1 \quad \underline{l}_2)^T \approx 0 \\ D\underline{m} &= (D_1 \quad D_2)(\underline{m}_1 \quad \underline{m}_2)^T \approx 0 \end{aligned} \tag{B.6}$$

where $D_1 = (1 \quad 1 \quad . \quad . \quad 1)^T$, $D_2 = \begin{pmatrix} x_1 & y_1 & x_1' & y_1' & x_1'' & y_1'' \\ x_2 & y_2 & x_2' & y_2' & x_2'' & y_2'' \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_n & y_n & x_n' & y_n' & x_n'' & y_n'' \end{pmatrix}$, $\underline{l}_1 = l_7$, $\underline{m}_1 = m_7$,

$\underline{l}_2 = (l_1 \quad l_2 \quad . \quad . \quad l_6)^T$, $\underline{m}_2 = (m_1 \quad m_2 \quad . \quad . \quad m_6)^T$ and n is the number of control points. We notice that the right-and side is equal to zero. This means that in our mixed LS-TLS solution above $B = \underline{0}$.

Following the algorithm above, we perform 1 (the number of error free columns in our data matrix D) householder transformation Q on our data matrix D to obtain:

$$Q^T(D_1 \quad D_2) = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \quad . \tag{B.7}$$

The Householder transformation Q of a vector $\underline{v}$ of length m is defined by:

$$Q = I_m - \frac{2\underline{u}\underline{u}^T}{\underline{u}^T\underline{u}} \quad , \tag{B.8}$$

where the vector $\underline{u}$ is given by:

$$\underline{u} = \underline{v} + \sigma\|\underline{v}\|e_1 \quad , \tag{B.9}$$

where $\sigma = \begin{cases} +1 \text{ if } \underline{v}_1 \geq 0 \\ -1 \text{ if } \underline{v}_1 < 0 \end{cases}$ and $e_1 = \begin{pmatrix} 1 \\ 0 \\ . \\ 0 \end{pmatrix}$.

We need to perform one Householder transformation Q on the data matrix D. This means the vector $\underline{v}$ (in (B.9) above) that is needed to find the matrix Q is equal to the first column of our data matrix, $\underline{v} = D_1 = (1 \quad 1 \quad . \quad 1)^T$. By substituting this vector into (B.9) above we are able to compute that the householder transformation matrix $Q = Q^T$ is equal to:

$$\frac{-1}{n+\sqrt{n}} \begin{pmatrix} 1+\sqrt{n} & 1+\sqrt{n} & 1+\sqrt{n} & . & . & 1+\sqrt{n} \\ 1+\sqrt{n} & 1-n-\sqrt{n} & 1 & . & . & 1 \\ 1+\sqrt{n} & 1 & 1-n-\sqrt{n} & & & 1 \\ . & . & & . & & . \\ . & . & & & . & . \\ 1+\sqrt{n} & 1 & 1 & . & . & 1-n-\sqrt{n} \end{pmatrix} \tag{B.10}$$

By multiplying this matrix, Q^T by the data matrix $D = (D_1 \quad D_2)$, as in (B.7) we are able to compute the matrices R_{11}, R_{12} and R_{22}.

The co-efficients l_1 to l_7 and m_1 to m_7 can then be found by solving the pairs of equations:

$$\begin{aligned} R_{11}\underline{l}_1 + R_{12}\underline{l}_2 &= 0 \\ R_{22}\underline{l}_2 &= \underline{\varepsilon} \end{aligned} \quad , \tag{B.11}$$

and

$$\begin{aligned} R_{11}\underline{m}_1 + R_{12}\underline{m}_2 &= 0 \\ R_{22}\underline{m}_2 &= \underline{\eta} \end{aligned} \quad . \tag{B.12}$$

The co-efficients $\underline{l}_2 = (l_1 \quad l_2 \quad . \quad . \quad l_6)^T$ and $\underline{m}_2 = (m_1 \quad m_2 \quad . \quad . \quad m_6)^T$ can be found by solving the equations:

$$\begin{aligned} R_{22}\underline{l}_2 &= \underline{\varepsilon} \\ R_{22}\underline{m}_2 &= \underline{\eta} \end{aligned} \quad , \tag{B.13}$$

as a total least squares problem. The solutions for the vectors $\underline{l}_2$ and $\underline{m}_2$ are the singular vectors of the matrix R_{22} corresponding to the two smallest singular values.

When the vectors $\underline{l}_2$ and $\underline{m}_2$ are known, we can find an expression for $\underline{l}_1 = l_7$ and $\underline{m}_1 = m_7$, by solving the equations:

$$\begin{aligned} R_{11}\underline{l}_1 + R_{12}\underline{l}_2 &= 0 \\ R_{11}\underline{m}_1 + R_{12}\underline{m}_2 &= 0 \end{aligned} \quad . \tag{B.14}$$

Substituting the matrix Q, in equation (B.10), into equation (B.7) gives:

$$\begin{aligned} R_{11} &= \left(\frac{-n\left(1+\sqrt{n}\right)}{n+\sqrt{n}} \right) \\ R_{12} &= \frac{\left(1+\sqrt{n}\right)}{n+\sqrt{n}} \left(\sum_{i=1}^{n} x_i \quad \sum_{i=1}^{n} y_i \quad \sum_{i=1}^{n} x_i' \quad \sum_{i=1}^{n} y_i' \quad \sum_{i=1}^{n} x_i'' \quad \sum_{i=1}^{n} y_i'' \right) \end{aligned} \quad . \tag{B.15}$$

On substituting these matrices into equation (B.14) and re-arranging we find that the expressions for l_7 and m_7 are given by:

$$\begin{aligned} l_7 &= \frac{1}{n}\sum_{i=1}^{n}\left(l_1 x_i + l_2 y_i + l_3 x_i' + l_4 y_i' + l_5 x_i'' + l_6 y_i''\right) \\ m_7 &= \frac{1}{n}\sum_{i=1}^{n}\left(m_1 x_i + m_2 y_i + m_3 x_i' + m_4 y_i' + m_5 x_i'' + m_6 y_i''\right) \end{aligned} \quad . \tag{B.16}$$

It can be seen that these expressions are the same as those obtained in section 4.2.1 as required.

Appendix C

Cholesky Decomposition

A symmetric positive definite matrix [Lüt96], A can be decomposed as:

$$A = LL^T \tag{C.1}$$

where L is an lower triangular matrix and can be thought of as the square root of A. This decomposition is known as the Cholesky decomposition [Lüt96, PTVF93]. This decomposition can also be written as:

$$A = R^T R \tag{C.2}$$

where R is an upper triangular matrix.

We can derive the matrix L by writing out (C.1) and equating both sides of the equation.

$$\begin{pmatrix} a_{11} & a_{12} & . & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & . & a_{2n} \\ a_{31} & a_{32} & . & . & . & a_{3n} \\ . & . & & . & & . \\ . & . & & & . & . \\ a_{n1} & a_{n2} & . & . & . & a_{n2} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 & . & . & 0 \\ l_{21} & l_{22} & 0 & . & . & 0 \\ l_{31} & l_{32} & l_{33} & & & . \\ . & . & . & . & . & . \\ . & . & . & & . & 0 \\ l_{n1} & l_{n2} & l_{n3} & . & . & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & l_{31} & . & . & l_{n1} \\ 0 & l_{22} & l_{32} & . & . & l_{n2} \\ 0 & 0 & l_{33} & . & . & l_{n3} \\ . & . & & . & & . \\ . & . & & & . & . \\ 0 & 0 & . & . & 0 & l_{nn} \end{pmatrix} \tag{C.3}$$

By equating entries of the matrices on both sides we obtain expressions for each l_{ji}:

$$l_{ii} = \sqrt{\left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)} \tag{C.4}$$

and

$$l_{ji} = \frac{1}{l_{ii}} \left(a_{ji} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right), \qquad i = i+1,..,n \quad . \tag{C.5}$$

Bibliography

[AHKO99] K Åström, A Heyden, F Kahl and Magnus Oskarsson, "Structure and Mation From Lines Under Affine Projections", In proc. Seventh International Conference on Computer Vision, pp 285-292, September 1999.

[Alo90] J Y Aloimonos, "Perspective approximations", Image and Vision Computing, 8(3), pp 179-192, 1990.

[Atk96] K. B. Atkinson, "Close range photogrammetry and machine vision", Whittles Publishing, 1996.

[AS97] S Avidan and A Shashua, "Novel View Synthesis in Tensor Space", In Proceedings of Computer Vision and Pattern Recognition Conference, pp 1034-1040, 1997.

[AS98] S Avidan and A Shashua, "Novel view synthesis by cascading trilinear tensors", IEEE Transactions on Visualization and Computer Graphics, Vol. 4, No. 4, pp 293-306 October – December 1998.

[Bas93] R Basri, "Recognition by Combinations of Model Views: Alignment and Invariance", In Proceedings Applications of Invariance in Computer Vision, Joint European-US Workshop, pp 435-450, Portugal, 1993.

[BBTS03] H H Baker, N Bhatti, D Tanguay, I Sobel, Dan Gelb, M E Goss, J MacCormick, W B Culbertson and T Malzbender, "Computation and Performance Issues in Coliseum, An Immersive Videoconferencing

System", In Proceedings of the Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA, pp 470-479, 2003.

[BCZ93] A Blake, R Curwen and A Zisserman, "Affine-invariant Contour Tracking with Automatic Control of Spatiotemporal Scale", In Proceedings of the 4th International Conference on Computer Vision, pp 66-75, 1993.

[Bey92] H. Beyer, "Accurate calibration of CCD-cameras", Proceedings of the International Conference of Computer Vision and Pattern Recognition, IEEE Computer Society Press, pp 96-101, Champaign, Illinois 1992.

[BL98] L Bretzner and T Lindeberg, "Use Your Hand as a 3-D Mouse, or, Relative Orientation from Extended Sequences of Sparse Point and Line Correspondences Using the Affine Trifocal Tensor", In Proceedings European Conference on Computer Vision, pp141-157, 1998.

[BM98] C Bregler and J Malik, "Tracking People with Twists and Exponential maps", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California, USA, pp 8-15, June, 1998.

[BSG98] B F Buxton, Z Shafi, and J Gilby, "Evaluation of the construction of novel views by a combination of basis views", in Signal Processing IX, Theories and Applications, Proceedings of EUSIPCO98, the IX European Signal Processing Conference, Greece, Vol III, pp 1285-1288, Eurasip September 1998.

[Bux01] B F Buxton, Private Communication, 2001.

[CRZ97] A Criminisi, I Reid and A Zisserman, "A Plane Measuring Device", In Proceedings, British Machine Vision Conference, 1997.

[dAHR98] L de Agapito, E Hayman and I Reid, "Self-Calibration of a Rotating Camera with Varying Intrinsic Parameters", In Proc. British Machine Vision Conference, pp 883-893, 1998.

[DB02] M B Dias and B F Buxton, "Integrated Shape and Pose Modelling", In Proc. British Machine Vision Conference (BMVC 2002), pp 827-836, September 2002.

[dBVOS97] M de Berg, M Van Kreveld, M Overmars and O Schwarzkopf, "Computational Geometry, Algorithms and Applications", Springer-Verlag, 1997.

[Dem87] J W Demmel, "The Smallest Perturbation of a Submatrix which Lowers the Rank and Constrained Total Least Squares Problems", SIAM Journal of Numerical Analysis, Vol. 24, No. 1, pp 199-206, 1987.

[DSB92] S Demey, A Zisserman and P Beardsley, "Affine and Projective Structure from Motion", In Proc. British Machine Vision Conference, pp 49-58, Leeds, 1992.

[DSSD99] X Decoret, G Schaufler, F Sillion and J Dorsey, "Multi-layered imposters for accelerated rendering", Eurographics, Vol. 18, No. 3, pp 145-156, 1999.

[DSTT00] F Dellaert, S Seitz, S Thrun and C Thorpe, "Feature Correspondence: A Markov Chain Monte Carlo Approach", In Advances in Neural Information Processing Systems (NIPS) 13, pp 852-858, 2000.

[DVOS97] M de Berg, M van Kreveld, M Overmars and O Schwarzkopf, "Computational Geometry, Algorithms and Applications" Springer-Verlag, 1997.

[Fau93] O Faugeras, "Three-Dimensional Computer Vision. A Geometric Viewpoint", MIT Press, 1993.

[Fau92] O D Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?" In Proc. European Conference on Computer Vision, LNCS 588, pp 563-578, Springer-Verlag, 1992.

[FL86] O D Faugeras and F Lustman, "Let us suppose the world is piecewise planar", International Symposium Robotics Research, pp 33-40, MIT press. 1986.

[FLM92] O D Faugeras, Q-T Luong and S J Maybank, "Camera Self-Calibration: Theory and Experiments", In Proc. European Conference on Computer Vision, LNCS 588, Springer-Verlag, pp 563-578, 1992.

[FM95a] O Faugeras and B Mourrain, "On the geometry and algebra of the point and line correspondences between *N* images", In Proceedings International Conference on Computer Vision, pp 951-956, 1995.

[FM95b] O Faugeras and B Mourrain, "About the correspondences of points between N images", In Proceedings of the IEEE Workshop on the Representation of Shapes, 1995.

[FRM98] S M Fairley, I D Reid and D W Murray, "Transfer of Fixation using Affine Structure: Extending the Analysis to Stereo", International Journal of Computer Vision, Volume 29, Issue 1, pp 47-58 1998.

[FRM95] S M Fairley, I D Reid and D W Murray, "Transfer of Fixation for an Active Stereo Platform via Affine Structure Recovery", In Proc. 5th International Conference on Computer Vision, Boston, pp 1100-1105, 1995.

[FVFH96] J D Foley, A Van Dam, S K Feiner and J F Hughs, "Computer Graphics Principles and Practice", Addison-Wesley Publishing Company, 1996.

[Gos86] A Goshtasby. "Piecewise Linear Mapping Functions for Image Registration", Pattern Recognition, 19(6), pp 459-466, 1986.

[GV80] G H Golub and C F Van Loan, "An Analysis of the Total Least Squares Problem", SIAM Journal of Numerical Analysis, Vol. 17, No. 6, pp 883-893, December 1980.

[GV96] G H Golub and C F Van Loan, "Matrix Computations", Third Edition, John Hopkins University Press. 1996.

[Han00] M E Hansard, Private Communication, 2000.

[Han99] M E Hansard, "Application of Multiview Techniques to the Visualisation of Historical Artefacts", Research Masters Thesis, Computer Vision, Image Processing, Graphics and Simulation. Supervised by Professor Bernard F. Buxton. University College London, 1999.

[Har92] R I Hartley, "Estimation of relative camera positions for uncalibrated cameras", In Proc. European Conference on Computer Vision, LNCS 588, pp 579-587, Springer-Verlag, 1992.

[Har94a] R Hartley, "Lines and Points in Three Views – An Integrated Approach", In Proceedings ARPA Image Understanding Workshop, 1994.

[Har94b] R Hartley, "Projective Reconstruction from Line Correspondences", IEEE Conference on Computer Vision and Pattern Recognition, pp 903-907, 1994.

[Har95a] R I Hartley, "A linear method for reconstruction from lines and points", In Proceedings International Conference on Computer Vision, pp 882-887, 1995.

[Har95b] R I Hartley, "Multilinear Relationships between Co-ordinates of Corresponding Image Points and Lines", In Proceedings of the International Workshop on Computer Vision and Applied Geometry, International Sophus Lie Center, Nordfjordeid, Norway, August, 1995.

[Har95c] R I Hartley, "In Defence of the 8-point Algorithm", In Proceedings of the International Conference on Computer Vision, pp 1064-1070, 1995.

[Har97a] R I Hartley, "Lines and Points in Three Views and the Trifocal Tensor", International Journal of Computer Vision 22(2), pp 125-140, 1997.

[Har97b] R I Hartley, "In Defense of the Eight-Point Algorithm", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 6, pp 580-593, 1997.

[Har98] R I Hartley, "Computation of the Quadrifocal tensor" In Proc. European Conference on Computer Vision, LNCS 1406, pp 20-35, Springer-Verlag, 1998.

[HB00a] M E Hansard and B F Buxton "Parametric View Synthesis", In Proceedings 6th European Conference on Computer Vision, pp 191-202, 2000.

[HB00b] M E Hansard and B F Buxton "Image-Based Rendering via the Standard Graphics Pipeline", In Proceedings IEEE International Conference on Multimedia and Exposition, pp 1437-1440, 2000.

[HC94] N Hollinghurst and R Cipolla, "Uncalibrated Stereo Hand-Eye Co-ordination", Image and Vision Computing, Volume 12, No. 3, pp 187-192, 1994

[Hey98] A Heyden, "A Common Framework for Multiple View Tensors", In Proceedings European Conference on Computer Vision, pp 3-19, 1998.

[Hey00] A Heyden, "Tensorial Properties of Multiple View Constraints", Mathematical Methods in the Applied Sciences, 23, pp169-202, 2000.

[HF89] T S Huang and O D Faugeras, "Some Properties of the E Matrix in Two-View Motion Estimation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(12), pp 1310-1312. 1989.

[HS79] B K P Horn and R W Sjoberg, "Calculating the Reflectance Map", Applied Optics, Vol. 18, No. 11, pp 1770-1779, 1979.

[HS88] C Harris and M Stephens, "A combined corner and edge detector", Forth Alvey Vision Conference, pp.147-151, 1988.

[HTM99] E Hayman, T Thórallsson and D W Murray, "Zoom-Invariant Tracking using Points and Lines in Affine Views", In Proc. Seventh International Conference on Computer Vision, pp 269-276, September 1999.

[HZ00] R Hartley and A Zisserman, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2000.

[ITKO04] F Isgrò, E Trucco, P Kauff and O Scheer, "3-D Image Processing in the Future of Immersive Media", IEEE Trans. on Circuits and Systems for Video Technology, Special issue on Immersive Telecommunications, Vol. 14, No. 3, pp 288-303, March 2004.

[Joh99] B Johansson, "View synthesis and 3D reconstruction of Piecewise Planar Scenes using Intersection Lines between the planes", in proceedings of the Seventh International Conference on Computer Vision, pp 54-59, September 1999.

[KB98] I Koufakis and B F Buxton, "Linear combination of face views for low bit rate face video compression", in Signal Processing IX, Theories and Applications, Proceedings of EUSIPCO98, the IX European Signal Processing Conference, Greece, Vol IV, pp 2305-2308, Eurasip, September 1998.

[KB99] I. Koufakis and B.F. Buxton, "Very low bit rate face video compression using linear combination of 2D face views and principal components analysis", Image And Vision Computing (17) 14, pp. 1031-1051. 1999.

[KBG99] D Kennedy, B F Buxton and J H Gilby, "Application of the Total Least Squares Procedure to Linear View Interpolation" in Proc BMVC'99, pp 305-314, 1999.

[KV91] J J Koenderink and A J van Doorn, "", Journal of the Optical Society of America, Vol. 8, No. 2, pp 377-385, 1991.

[KV96] J J Koenderink and A J van Doorn, "Bidirectional Reflection Distribution Function expressed in terms of scattering modes", In Proceedings of European Conference on Computer Vision, pp 28-39, 1996.

[LAO02] M I A Lourakis, A A Argyros and S C Orphanoudakis, "Detecting Planes in an Uncalibrated Image Pair", In Proceedings of British Machine Vision Conference, pp 587-596, Cardiff, 2002.

[LDFP93] Q-T Luong, R Deriche, O D Faugeras and T Papadopoulo, "On Determining the Fundamental Matrix: Analysis of Different Methods and Experimental Results", INRIA Research Report No 1894, 1993.

[Len98] J. Lengyel, "The Convergence of Graphics and Vision" IEEE Computer, Vol. 31, Issue 7, pp 46-53, July 1998.

[LF96] Q-T Luong and O Faugeras, "The Fundamental Matrix: Theory, Algorithms and Stability Analysis", International Journal of Computer Vision, Vol. 17, pp 43-75, 1996.

[L-H81] H C Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from Two Projections", Nature, 293(10), pp 133-135. 1981.

[LH95] C L Lawson and R J Hanson, "Solving Least Squares Problems", Classics in Applied Mathematics, SIAM. 1995.

[LHO00] M I A Lourakis, S T Halkidis and S C Orphanoudakis, "Matching Disparate Views of Planar Surfaces using Projective Invariants", Image and Vision Computing, 18, pp 673-683, 2000.

[Lip91] S Lipschutz, " Linear Algebra", Schaum's Outlines. McGraw-Hill, 1991.

[LTAO00] M I A Lourakis, S V Tzurbakis, A A Argyros and S C Orphanoudakis, "Using Geometric Constraints for Matching Disparate Stereo Views of 3D Scenes containing Planes", In Proceedings International Conference on Pattern Recognition (ICPR'00), Barcelona, Spain, pp 419-422, September 2000.

[Lüt96] H Lütkepohl, "Handbook of Matrices", John Wiley & Sons Ltd. 1996.

[MB95] L McMillan and G Bishop, "Plenoptic Modelling: An Image-Based Rendering System", In Proceedings of SIGGRAPH, Los Angeles, pp 39-46, August 1995.

[MC98] P R S Mendonça and R Cipolla R "Analysis and Computation of an Affine Trifocal Tensor", In Proceedings British Machine Vision Conference, BMVC'98, pp 125-133, 1998.

[MM98] M Mühlich and R Mester, "The Role of Total Least Squares in Motion Analysis", In Proceedings of 5th European conference on Computer Vision, ECCV'98, Springer Verlag, pp 305-321, 1998.

[MZ92] J L Mundy and A Zisserman, "Geometric Invariance in Computer Vision", MIT Press, 1992.

[MZKD04] J Mulligan, X Zabulis, N Kelshikar and K Daniilidis, "Stereo-based Environment Scanning for Immersive Telepresence", IEEE Trans. on Circuits and Systems for Video Technology, Special issue on Immersive Telecommunications, Vol. 14, No. 3, pp 304-320, March 2004.

[Peb98] P P Pebay, "Consruction d'une constrainte Delaunay-admissible en dimension 2", INRIA Research Report No 3492, September 1998.

[PH99] S Pollard and S Hayes, "View synthesis by edge transfer with application to the generation of immersive video objects", In proceedings of the ACM Symposium on Virtual Reality Software and Technology, Taipei, Taiwan, pp 91-98, 1999.

[PHPL97] S Pollard, S Hayes, M Pilu and A Lorusso, "Automatically Synthesising Virtual Viewpoints by Trinocular Image Interpolation", HP Laboratories Technical Report, HPL-97-166, December 1997.

[Pil97] M Pilu, "A Direct Method for Stereo Correspondence based on Singular Value Decomposition", In Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 261-266, 1997.

[PK94] C J Poelman and T Kanade, "A Para-perspective Factorization Method for Shape and Motion Recovery", In Proceedings Third European Conference on Computer Vision, ECCV'94, pp 97-108. 1994.

[PKVV98] M Pollefeys, R Koch, M Vergauwen and L Van Gool, "Metric 3D Surface Reconstruction from Uncalibrated Image Sequences", In Proceedings SMILE (3D Structure from Muliple Images of Large-Scale Environments) European Workshop, pp139-154, 1998.

[PPHL98] S Pollard, M Pilu, S Hayes and A Lorusso, "View synthesis by trinocular edge matching and transfer", In proceedings of the British Machine Vision Conference 1998.

[Pra91] W K Pratt, "Digital Image Processing", A WileyInterscience Publication, 1991.

[PTVF93] W H Press, S A Teukolsky, W T Vetterling, B P Flannery, "Numerical Recipes in C", Second Edition, Cambridge University Press, 1993.

[QK96] L Quan and T Kanade, "A Factorization Method for Affine Structure from Line Correspondences", In proc. 15th Computer Vision and Pattern Recognition, pp 803-808, June 1996.

[SA90] M Spetsakis and J Aloimonos, "A Unified Theory of Structure from Motion", In Proceedings DAPRA Image Understanding Workshop, pp 271-283, 1990.

[SA91] M Spetsakis and J Aloimonos, "A Mulit-frame Approach to Visual Motion Perception", International Journal of Computer Vision, Vol. 6, No. 3, pp 245-255, 1991.

[SA00] A Shashua and S Avidan, "On the Reprojection of 3D and 2D Scenes Without Explicit Model Selection", In Proceedings European Conference on Computer Vision, pp 936-949, 2000.

[SD95] S M Seitz and C R Dyer, "Physically-valid view synthesis by image interpolation", IEEE Workshop on Representation of Visual Scenes, pp 18-25, 1995.

[SD96] S M Seitz and C R Dyer, "View Morphing", In Proc. SIGGRAPH, pp 21-30, 1996.

[SFZ99] R A Smith, A W Fitzgibbon and A Zisserman, "Improving Augmented Reality using Image and Scene Constraints", In Proc. British Machine Vision Conference, Nottingham, pp 295-304, September 1999.

[Sha92] A Shashua, "Geometry and Photometry in 3D Visual Recognition", PhD Thesis, Massachusetts Institute Of Technology, 1992.

[Sha94] A Shashua, "Algebraic Functions for Recognition" A.I. Memo No. 1492, C.B.C.L. Paper No. 90, Massachusetts Institute of Technology Artificial Intelligence Laboratory and Center for Biological Computational Learning Whitaker College, 1994.

[Sha95] L S Shapiro, "Affine Analysis of Image Sequences", Cambridge University Press, 1995.

[SHB99] M Sonka, V Hlavac and R Boyle, "Image Processing, Analysis, and Machine Vision", Second Edition, Brooks/Cole Publishing Company, 1999.

[SK98] J G Semple and G T Kneebone, "Algebraic Projective Geometry" Oxford Classic Texts. ISBN 0198503636. 1998.

[SS99] H-Y Shum and R Szeliski, "Stereo reconstruction from multiperspective panoramas", In Proceedings of the Seventh International Conference on Computer Vision, Kerkyra, Greece, pp 14-21, September 1999.

[SW00] A Shashua and L Wolf, "On the Structure and Properties of the Quadrifocal Tensor", In Proceedings European Conference on Computer Vision, pp 710-724, 2000.

[SZB95] L S Shapiro, A Zisserman and M Brady, "3D Motion Recovery via Affine Epipolar Geometry", International Journal of Computer Vision, 16, pp 147-182. 1995.

[TCD01] T Hawkins, J Cohen and P Debevec, "A Photometric Approach to Digitizing Cultural Artifacts", In Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage, Glyfada, Greece, pp 333-342, 2001.

[TK92] C Tomasi and T Kanade, "Shape and Motion from Image Streams under Orthography: a Factorization Method", International Journal of Computer Vision, 9(2), pp 137-154, 1992.

[TM99] T Thórhallsson and D W Murray, "The Tensors of Three Affine Views" In Proceedings Computer Vision and Pattern Recognition Conference, pp 1450-1456, 1999.

[Tri95] B Triggs, "Matching Constraints and the Joint Image", In Proceedings International Conference on Computer Vision, pp 338-343, 1995.

[Tri96] B Triggs, "Linear Projective Reconstruction from Matching Tensors", In Proceedings British Machine Vision Conference, pp 665-674, 1996.

[Tsa86] R. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision", In Proceedings. Computer Vision and Pattern Recognition Conference, pp 364-374, 1986.

[TV98] E Trucco and A Verri, "Introductory Techniques for 3D Computer Vision", Prentice Hall. 1998.

[TZ96] P H S Torr and A Zisserman A, "Robust Parameterization and Computation of the Trifocal Tensor", In Proceedings British Machine Vision Conference, pp 655-664, 1996.

[TZ97] P H S Torr and A Zisserman A, "Robust Parameterization and Computation of the Trifocal Tensor", Image and Vision Computing 15, pp 591-605. 1997.

[UB91] S Ullman and R Basri, "Recognition by Linear Combination of Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(10), pp 992-1006, 1991.

[Ull96] S Ullman, "High-level Vision, Object Recognition and Visual Cognition", MIT Press, 1996.

[VV89] S Van Huffel and J Vandewalle, "Analysis and properties of the generalized total least squares problem AX≅B when some or all columns in A are subject to error", SIAM Journal of Matrix Analysis and Applications, 10(3), pp 294-315, 1989.

[VV91] S Van Huffel and J Vandewalle, "The Total Least Squares Problem, Computational Aspects and Analysis", SIAM Frontiers in applied Mathematics 9, 1991.

[WHA92] J Weng, T Huang and N Ahuja, "Motion and Structure from Line Correspondences: Closed-Form Solution, Uniqueness and

Optimization", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 3, pp 318-336, 1992.

[Wil65] J H Wilkinson, "The Algebraic Eigenvalue Problem", Clarendon, 1965.

[WP98] A Watt and F Policarpo, "The Computer Image", Addison Wesley, 1998.

[WW92] A Watt and M Watt, "Advanced Animation and Rendering Techniques, Theory and Practice", Addison Wesley, 1992.

[YM98] Y Yu and J Malik, "Recovering Photometric Properties of Architectural Scenes from Photographs", In proceedings of 25th International Conference on Computer Graphics and Interactive Techniques, pp 207-217, 1998.

[Zha98] Z Zhang, "Image-Based Geometrically-Correct Photorealistic Scene/Object Modeling (IBPhM): A Review", Invited Talk, In Proceedings of the Asian Conference on Computer Vision (ACCV), pp 340-349, 1998.

[Zis92] Zisserman A. "Notes on Geometric Invariance in Vision". British Machine Vision Conference, Tutorial. 1992.